DEPARTMENT OF DEFENCE

DEFENCE SCIENCE & TECHNOLOGY ORGANISATION

DSTO

# Experiences in the Development of EXC3ITE-Based HLA-Compliant Simulation Capabilities

Mike Davies, Carsten Gabrisch, Karyn Matthews and John M. Dunn

DSTO-TR-1147

# Experiences in the Development of EXC3ITE-Based HLA-Compliant Simulation Capabilities

*Mike Davies, Carsten Gabrisch, Karyn Matthews, John M. Dunn*

**Information Technology Division**
**Electronics and Surveillance Research Laboratory**

DSTO-TR-1147

## ABSTRACT

Program Takari's Experimental C3I Technology Environment (EXC3ITE) will be key to the establishment of modelling and simulation architectures, practices and capabilities that are integrated with real C3I systems to support a range of military experimentation. The US High Level Architecture (HLA) is one component of an emerging encapsulation and architectural standard for simulation that has been mandated within the US DoD. EXC3ITE and the HLA share two main principles of interoperability and reuse; they are both middleware-rich. This commonality needs to be exploited in order to make best use of the HLA standard. EXC3ITE-based HLA-compliant simulation capabilities were recently developed employing a combination of Adacel Technologies and Aspect Computing staff. This report describes the capabilities developed and experiences gained and raises issues and recommendations on the way ahead for EXC3ITE-based simulation and synthetic environments.

**Approved for Public Release**

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

*Published by*

*DSTO Electronics and Surveillance Research Laboratory*
*PO Box 1500*
*Salisbury South Australia 5108 Australia*

*Telephone: (08) 8259 5555*
*Fax: (08) 8259 6567*

*AR-011-860*
*May 2001*

**APPROVED FOR PUBLIC RELEASE**

# Experiences in the Development of EXC3ITE-Based HLA-Compliant Simulation Capabilities

## Executive Summary


Program Takari's Experimental C3I Technology Environment (EXC3ITE) will be key to the establishment of modelling and simulation architectures, practices and capabilities that are integrated with real C3I systems to support a range of military experimentation. The US High Level Architecture (HLA) is one component of an emerging encapsulation and architectural standard for simulation that has been mandated within the US DoD. The US is particularly reliant on HLA providing a means of achieving interoperability between simulation and real-world C3I systems. EXC3ITE and the HLA share two main principles of interoperability and reuse; they are both middleware-rich. This commonality needs to be exploited in order to make best use of the HLA standard. EXC3ITE-based HLA-compliant simulation capabilities were recently developed employing a combination of Adacel Technologies and Aspect Computing staff.

The following products were developed and migrated to the EXC3ITE demonstration environment:

- An HLA-enabled version of the existing Distributed Interactive C3I Effectiveness (DICE) simulation;
- The DICE Track Service capability to provide synthetic tracks as part of the EXC3ITE Track Services and according to the associated EXC3ITE standard;
- An HLA Track Federate enabling the EXC3ITE Track Service to be used as a federate within an HLA-compliant simulation federation.

The project culminated with a demonstration to the Takari Executive on 29 June 2000 that illustrated concepts and capabilities for building a modular distributed synthetic environment for the purpose of addressing a particular military application. An overarching theme of Situation Awareness & Planning provided context to the demonstration of:

- A hybrid of simulation and EXC3ITE federates within an HLA federation;
- A presence of simulated C2 with HLA using DICE;
- A level of synergy and reuse between HLA and EXC3ITE through the EXC3ITE Track Services;
- An ability to use in-country plus international HLA object models;

- A hybrid of HLA and other standards and approaches including use of current DICE interoperability capabilities; and

- Distribution of the synthetic environment between Canberra and Salisbury DSTO sites.

The project was a pathfinder for EXC3ITE-based simulation and synthetic environments and provides a foundation for further R&D. The project illustrated that interoperability and reuse are achievable between the architecture and practices aspired to for real C3I systems and that of simulation and synthetic environments. The existence of a standard language within DICE permits its participation in a range of HLA exercises with minimal software coding. The deliverables from this project form important elements of ITD's Joint Synthetic Environment Facility (JOSEF) and aid exploration of the synthetic environment element of an integrated modelling environment for operational planning.

This report describes the capabilities developed and experiences gained and raises issues and recommendations on the way ahead for EXC3ITE-based simulation and synthetic environments. It is important to acknowledge that HLA alone is not a panacea. HLA can only be successfully adopted in conjunction with pursuing other initiatives of a modelling and simulation *common technical framework*. Data standards and conceptual models that convey a common view and language are comparable with the goals of the EXC3ITE Track Services and Geospatial Services Segment (GSS) focus groups, and EXC3ITE ideals in general. In the case of EXC3ITE Track Services, the project showed that common data standards and conceptual models can be identified between simulation and real systems. EXC3ITE offers the potential for common software and tools, and resource repositories for simulation.

The project demonstrated the use of HLA to convey orders, reports and requests, or *C2 messages*, modelled explicitly by the DICE simulation. Investigation is needed into the means by which HLA can enable C3I systems and agencies to become battlefield entities that can be subjected to targeting, degradation and destruction by physical or non-physical means. It is felt that HLA object models and interactions based on formatted textual messages should adequately address C2 language issues and that the DICE HLA development offers a significant capability that will aid interoperation with real C3I systems. C2 object model standardisation is needed that maximises alignment with real C3I system developments and works towards a 'plug and play' capability using HLA. Capabilities that enable real and prototype C3I systems to interoperate with simulated systems are critical and this is a key reason for maximising a relationship between EXC3ITE, simulations and synthetic environments.

The project has established a foundation upon which such issues can be explored. A TTCP focus area on "Simulation-C4ISR Interoperability" is intended to further explore the significance of HLA to C2.

The baseline EXC3ITE simulation capability provided by this project needs to be maintained and extended.

# Authors


## Mike Davies
Information Technology Division


*Mike Davies' career at the DSTO has involved the application of mathematical modelling and computer simulation to the analysis of human performance in ground-based air defence systems and aircraft navigation and weapon delivery. As Head of Systems Simulation and Assessment Group in Information Technology Division his current interests concern the modelling, simulation and assessment of military Command and Control and the provision of tools and synthetic environments for course-of-action analysis, mission rehearsal and training.*


## Carsten Gabrisch
Information Technology Division


*Carsten Gabrisch is a professional officer with a BSc in applied mathematics. His career at the DSTO has involved the application of mathematical modelling and computer simulation to ground-based air defence systems and the development of a deployment tool for ground-based air defence systems. Currently he is involved in the development of simulation tools to study the effectiveness of Command, Control, Communication and Intelligence.*


## Karyn Matthews
Information Technology Division


*Karyn obtained a BSc(Hons) in Applied Mathematics and a PhD in Computational Oceanography at the University of Adelaide. Her career at the DSTO has involved the application of mathematical modelling and computer simulation to the analysis of missile flight dynamics, blast characterisation of warheads, and evaluation and enhancement of military Command, Control, Communication and Intelligence systems. Her main current interest involves using simulation and synthetic environments for experimentation with military command and control.*


## John M. Dunn
Information Technology Division


*John is a professional officer with a BAppSc in Mathematics and Computing, and is currently pursuing a GradCert in Operations Research and Statistics. His career at DSTO has involved the application of computer simulation to weapons platforms and military Command, Control, Communication and Intelligence systems. Currently he is involved in the development and application of computer simulation, modelling and analysis techniques in the C3I domain.*

# Contents

# Abbreviations

| | |
|---|---|
| ACT | Australian Capital Territory |
| ADF | Australian Defence Force |
| ADO | Australian Defence Organisation |
| AEW&C | Airborne Early Warning and Control |
| API | Application Programming Interface(s) |
| AVT | Asset Visualisation Tool |
| C2 | Command and control |
| C3I | Command, Control, Communication and Intelligence |
| C4ISR | Command, Control, Communication, Computers, Intelligence, Surveillance and Reconnaisance |
| CAP | Combat Air Patrol(s) |
| CCSIL | Command and Control Simulation Interface Language |
| CMMS | Conceptual Models of the Mission Space |
| CNF | Common Node Framework |
| COA | Course(s) of Action |
| CORBA | Common Object Request Broker Architecture |
| CSS | Command Support System(s) |
| DICE | Distributed Interactive C3I Effectiveness |
| DIS | Distributed Interactive Simulation |
| DMDD | DICE Message Definition Database |
| DMSO | Defense Modelling and Simulation Office |
| DoD | Department of Defence |
| DSTO | Defence Science and Technology Organisation |
| EXC3ITE | Experimental C3I Technology Environment |
| FEDEP | Federation Development and Execution Process |
| FOM | Federation Object Model |
| GIS | Geospatial Information System(s) |
| GSS | Geospatial Services Segment |
| HLA | High Level Architecture |
| IDL | Interface Definition Language |

| | |
|---|---|
| IEEE | Institute of Electrical and Electronic Engineers |
| INE | Integrated Natural Environment |
| ITD | Information Technology Division |
| JOSEF | Joint Synthetic Environment Facility |
| JSA | Joint Systems Analysis |
| JTLS | Joint Theater Level Simulation |
| LOD | Land Operations Division |
| LSA | Land Situation Awareness |
| M&S | Modelling and simulation |
| MRCI | Modular Reconfigurable C4I Interface |
| NATO | Northern Atlantic Treaty Organization |
| OGC | Open GIS Consortium |
| OMG | Object Management Group |
| OMT | Object Model Template |
| PDS | Phoenix Display System |
| PDU | Protocol Data Unit(s) |
| PUI | Peripheral Unit Interface |
| R&D | Research and development |
| RDBMS | Relational Database Management System |
| RPR-FOM | Real-time Platform-level Reference FOM |
| RTI | Run-Time Infrastructure(s) |
| SEDRIS | Synthetic Environment Data Representation and Interchange Specification |
| SOM | Simulation Object Model |
| SSD | Surveillance Systems Division |
| TAO | The ACE ORB |
| TDRAP | Technology Demonstrator for the Recognised Air Picture |
| TTCP | The Technical Cooperation Program |

# 1. Introduction

The Experimental C3I Technology Environment (EXC3ITE) is an environment within which concepts and technology for future C3I capabilities can be explored and developed. EXC3ITE is the integrator for the C3I R&D undertaken in DSTO's Takari R&D program. Takari's EXC3ITE environment will be key to the establishment of modelling and simulation (M&S) architectures, practices and capabilities that are integrated with real C3I systems to support a range of military experimentation.

In order to realise the benefits of using M&S in support of warfare functions such as situation awareness, planning and mission rehearsal and execution, there is a need to maximise the timely reuse, synthesis and interoperability of disparate models and simulations. The US High Level Architecture (HLA) is one component of an emerging encapsulation and architectural standard for simulation that has been mandated within the US DoD. The US is particularly reliant on HLA providing a means of achieving interoperability between simulation and real-world C3I systems. The Australian Defence Organisation (ADO), that includes the DSTO, needs to explore and develop HLA capabilities in the interest of Australian Joint force and individual Service M&S plus issues concerning coalition operations.

EXC3ITE and the HLA share two main principles of interoperability and reuse; they are both middleware-rich. This commonality needs to be exploited in order to make best use of the HLA standard. EXC3ITE-based HLA-compliant simulation capabilities were recently developed employing a combination of Adacel Technologies and Aspect Computing staff. There is a need to investigate standards, architectures and practices for EXC3ITE-based simulation plus strategies for enabling technical and managerial leverage between EXC3ITE and simulation (particularly HLA-compliant) applications.

# 2. Overview of EXC3ITE

Reference 1 describes EXC3ITE as 'an enabling environment for experimenting with new technology and work practices to assist the ADF in improving its command and evolving an enhanced C4ISR capability'. Moreover, 'the goal of EXC3ITE is to demonstrate how it is possible to build efficiently and effectively, a future-proof integrated C3I system by using an architecture-based approach.' Modelling and simulation will play a vital role. A model integrated C3I system is being established through a distributed object-oriented component-based environment partitioned as shown in Table 1 to aid the transition from R&D to military capability. New ideas for R&D services and components are trialled within the EXC3ITE prototype environment at the appropriate security classification level. When stable such products are transitioned to the appropriate EXC3ITE demonstration environment where they become supported enabling user experimentation and evaluation.

| | |
|---|---|
| Prototype (Restricted) | Prototype (Secret) |
| Demonstration (Restricted) | Demonstration (Secret) |

**Table 1: EXC3ITE environments**

Two EXC3ITE services are particularly pertinent to this report; namely Track Services and the Geospatial Services Segment (GSS). An overview of these services follows.

## 2.1 EXC3ITE Track Services

An EXC3ITE focus group for track services exists and is made up of representatives from multiple DSTO divisions. The aim of the group is 'to design, implement and demonstrate CORBA compliant interfaces that are applicable for any repository of target track data. Such interfaces will enable a repository of track data to present itself within EXC3ITE as a track service while hiding the internal details of that repository' [2,3]. A client of target track data is able to pull information as required from a track service.

The EXC3ITE track service specification is based on the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) and its Common Object Services, adopting relevant international standards where appropriate, and using the OMG Interface Definition Language (IDL)[4]. The IDL is a language independent way of specifying the interfaces to CORBA services that can be mapped automatically to several other languages for implementation. Given that target track data contains geospatial properties, compliance with Open GIS Consortium (OGC) Inc. standards such as the *OpenGIS Simple Features Specification for CORBA* are pursued. It is aimed to achieve compliance of temporal properties with the *OMG CORBA Time Service Specification* standard.

The following definitions apply[4]:.

- A *Track Point* denotes an updated estimate of the state of a particular target.
- A *Track* denotes a chronological sequence of track points pertaining to the same target.

There are currently three DSTO R&D systems that handle Track Services track data, namely:

- Information Technology Division's (ITD) Distributed Interactive C3I Effectiveness (DICE) simulation software suite (provides simulated track data)[10];
- Land Operations Division's (LOD) Land Situation Awareness (LSA)-C4ISR (provides simulated track data); and

- Surveillance Systems Division's (SSD) Technology Demonstrator for the Recognised Air Picture (TDRAP) (provides real track data).

The DICE track service receives particular attention in this report since it forms a key vehicle for the HLA/EXC3ITE project. It is important to stress, however, that the HLA-compliance track service federate achieved as part of the project can be used with any track service compliant with the IDL. The DICE track service is detailed in Appendix A; DICE itself is discussed further in later sections.

## 2.2 EXC3ITE Geospatial Services Segment

The Geospatial Services Segment (GSS) of EXC3ITE aims to define interfaces and access methods to geospatial data (geodata) and geoprocesses within a framework able to efficiently support military operations and associated C2[5]. The EXC3ITE GSS seeks to provide a service based on COTS and custom-developed applications having unified access to geodata and geoprocessing software components from a variety of sources and employing existing open standards where possible.

Building blocks of the GSS include:

- Resource discovery tools that enable quick discovery of and access to the geodata and products needed to satisfy the requirements for a particular task;

- The geodata repository which is the core component of the GSS and a logical entity whose implementation can be distributed across the EXC3ITE network; and

- Geoprocessing software components that provide applications with a common interface to services such as route planning and line of sight calculations.

A key issue in the provision of EXC3ITE-based simulations and synthetic environments is the relationship required between GSS and the Synthetic Environment Data Representation and Interchange Specification (SEDRIS) standard and tools. The objectives of the SEDRIS program are to[9]:

- Articulate and capture the complete set of data elements and associated relationships needed to fully represent the physical environment;

- Support the full range of simulation applications across all environmental domains (terrain, ocean, atmosphere, and space);

- Provide a standard interchange mechanism to pre-distribute environmental data (from primary source data providers and existing resource repositories) and promote data base reuse and interoperability among heterogeneous simulations.

# 3. Overview of HLA

It is important to be aware of the context within which the High Level Architecture (HLA) was initiated. The 1995 US Department of Defence M&S Master Plan conceived the notion of a *common technical framework* that remains embraced within current thrusts of the US Defense Modelling and Simulation Office (DMSO)[6,7]. A common technical framework encompasses:

- Common representations of data across models, simulations and C3I systems via *data standardisation;*
- A common world view via *Conceptual Models of the Mission Space (CMMS)* that provide simulation-independent conceptual descriptions of real-world processes, entities, environments, and relationships; and
- A *high level architecture* for simulation.

The goal of a common technical framework is complemented by:

- Authoritative *representations of human behavior* (individual and group);
- An *Integrated Natural Environment (INE)* program that provides authoritative representations of atmosphere, space, oceans, and terrain (includes SEDRIS); and
- Common software and tools, help desks, education, and M&S resource repositories.

HLA is therefore just one part of an essential integrated set of capabilities and infrastructure. It is important to recognise this when pursuing concepts and capabilities for EXC3ITE-based simulation and synthetic environments (discussed later).

HLA is a general-purpose architecture for simulation interoperability and reuse. It represents a composable approach to constructing an interacting set (a *federation*) of simulations (*federates*) that conform to an interface standard. An individual simulation or set of simulations developed for one purpose can be applied to another application under the HLA concept. Significant reuse ultimately reduces the cost and time required to create a federation for a new purpose, and fosters the possibility of distributed collaborative development of complex simulation applications[7].

The functionality of individual federates of the federation are separated from the underlying general-purpose simulation infrastructure, which is provided by a Run-Time Infrastructure (RTI), as shown in Figure 1. A RTI is freely available from the DMSO; in addition, a number of commercial RTI implementations exist or are currently under development. Links with external systems (such as C3I systems and live participants) present the same kind of interface to the RTI as any other federate; the external interface to a particular system is essentially a software wrapper for that system.

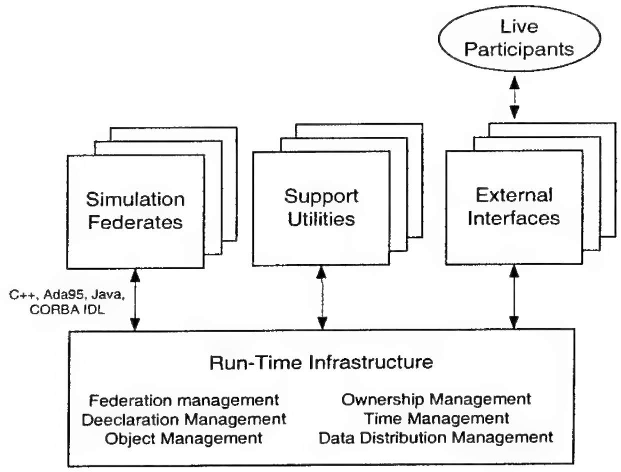


**Figure 1 HLA configuration**

HLA itself is composed of three interrelated elements:

- The HLA Object Model Template (OMT) that provides a common means of describing the functionality and language of communication of federates and federations;

  *The OMT is the standard structural framework for specifying HLA object models of individual member federates and federations. A federate's Simulation Object Model (SOM) is a description of data required and provided by the federate. The Federation Object Model (FOM) is the result of a manual melding of individual SOM, resolving any conflict, and is used to specify the nature of the data exchange required among federates within a specific federation. The OMT is discussed further in Appendix B.*

- The HLA Interface Specification that defines how federates interact with the RTI and its services;

  *The HLA interface specification is an Institute of Electrical and Electronic Engineers (IEEE) standard 1516. V1.3 of this specification was recently formally adopted by the OMG as an industry standard for open distributed computing[8]. It is a generic specification for the various language-specific Application Programming Interfaces (API) that may be employed between any component federate and the RTI services. RTI services fall into six categories: Federation Management, Declaration Management, Object Management, Ownership Management, Time Management, and Data Distribution Management (Appendix B). New HLA services not available in DIS include better Time Management, Declaration Management, and Ownership Management. The broadcast communications of DIS are replaced with multicasting.*

- HLA Rules governing compliance with the overall architecture.

  *HLA compliance applies to federates, federations and RTI. A checklist for federates and federations is discussed in Appendix B. There also exists a compliance checklist for RTI which needs to be adhered to if one is developing an RTI.*

It is important to note that HLA is an architectural standard that promotes appropriate packaging of simulations in order to facilitate interoperability and reuse. Achievement of interoperability and reuse within a range of federations necessitates standardisation of object models and/or flexibility to deal with a range of object models. The term *HLA-enabled* will be used to convey a simulation that has the general mechanics to enable connection to a HLA federation. This is in contrast to a simulation that is *HLA-compliant* and able to interoperate within the context of a chosen federation.

The Federation Development and Execution Process (FEDEP) Model describes a high-level framework for the development and execution of HLA federations. The intent of the FEDEP Model is to specify a set of guidelines for federation development and execution that federation developers can leverage to achieve the needs of their application. The six basic steps of the FEDEP are:

- Step 1: Define Federation Objectives.
- Step 2: Develop Federation Conceptual Model ie an appropriate representation of the real world domain.
- Step 3: Design Federation by determining federates and assigning functional responsibilities.
- Step 4: Develop Federation by defining the FOM, establishing federate agreements on consistent databases and algorithms, and modifying federates as required.
- Step 5: Integrate and Test Federation, particularly to ensure interoperability.
- Step 6: Execute Federation and Prepare Results.

For the HLA/EXC3ITE project the process was not as linear as suggested above and was heavily influenced by available products and project constraints.

# 4. Aims and Deliverables of the HLA/EXC3ITE Project

The format of this section is aligned with Steps 1 and 2 of the FEDEP.

## 4.1 Project and federation objectives and strategy

The aims of the HLA/EXC3ITE project were to:

- Investigate the significance to C2 of adopting HLA as a standard;
- Determine strategies needed to enable leverage between EXC3ITE and HLA applications; and

- Demonstrate capability, specifically:
    - A hybrid of M&S federates and EXC3ITE federates;
    - A level of synergy and reuse between HLA and EXC3ITE;
    - A presence of C2 simulation with HLA;
    - Reuse of own SOM and software libraries;
    - An ability to adopt an existing international SOM; and
    - A hybrid of HLA and other 'standards'.

The DICE simulation software suite[10] enables explicit simulation of C3I functions, processes and systems and their interaction with the physical domain of platforms, sensors etc. DICE is currently able to interoperate with a range of disparate models and simulations using tailored or standard (such as DIS) interfaces and protocols. It was considered an ideal tool for exploring HLA and C2 (expanded upon in later discussions) through development of an HLA-enabled version and demonstration of its ability to interoperate with other chosen HLA-compliant simulations within a federation. Such a federation could be demonstrated on the EXC3ITE network. Including selected EXC3ITE components and services to form a hybrid federation would bring the capability within the EXC3ITE environment. The hybrid federation would enable issues concerning a melding of HLA with the EXC3ITE technical framework and software environment to be identified and addressed. The hybrid federation could be used to demonstrate concepts and capabilities for EXC3ITE simulation and synthetic environment services able to drive real-world command support systems (CSS) plus a range of EXC3ITE-based experimental and prototype technologies.

The deliverables from the project were therefore:

- Migration to the EXC3ITE Demonstration Environment of:
    - An HLA-enabled version of the DICE simulation;
    - The DICE Track Service capability able to provide synthetic tracks as part of the EXC3ITE Track Services and according to the associated EXC3ITE standard;
    - An HLA Track Federate that enables EXC3ITE Track Services to be a federate within an HLA-compliant simulation federation.
- A demonstrable HLA-based federation.

The general strategy was therefore to:

- Develop an HLA-enabled version of the DICE simulation;

- Design a simulation federation with a defined purpose;
- Select one or more other simulation federates to participate in the federation; make federates HLA compliant if needed and achievable within project constraints;
- Select a particular RTI that is HLA compliant;
- Develop an ability to run the simulation federation on the EXC3ITE network using specific simulation middleware and software components;
- Design a federation that uses a combination of simulation federates plus selected EXC3ITE components or services having a defined purpose; maximise reuse of information and software associated with the EXC3ITE 'federates';
- Develop an ability to run the HLA/EXC3ITE federate within the EXC3ITE environment, exploring issues concerning a melding with and reuse of the EXC3ITE software environment;
- Demonstrate developments in the context of a chosen military application;
- Abstract and explore general issues concerning HLA and EXC3ITE, using specific products and application areas as vehicles; and
- Make recommendations concerning the way ahead.

## 4.2 Conceptual model of required federation

The required federation, as conceptualised at the start of the HLA/EXC3ITE project, is conveyed pictorially as shown in Figure 2. It captures the essence of general requirements but is shaped by an awareness of the specific products available to the project. Within the context of a Air/Land operation, the federation would provide a synthetic environment able to stimulate an audience of military planners and their command support systems (CSS), and be able to respond to resultant experimental tasking from those planners. The outlook was as follows:

- Sensor models within a physical domain simulation, coupled with models of other intelligence sources, sense activities in the military geographic region,
- Reports of which are passed through a simulated reporting chain to explicitly model information flow and associated delay.
- The simulated reports are combined with real track data feeds and stimulate the support systems of operational planners (not addressed in the eventual developments; this melding of real and simulated data has significant management issues).
- In order to explore Course of Action (COA) development and analysis issues, the planners experiment with the tasking of various physical assets.

- Tasking is issued through a simulated command chain and COA activities played out within a physical domain simulation and visualised in a 3D geospatial environment fed by position reports from the tasked assets.

- The loop is closed through the sensor models.

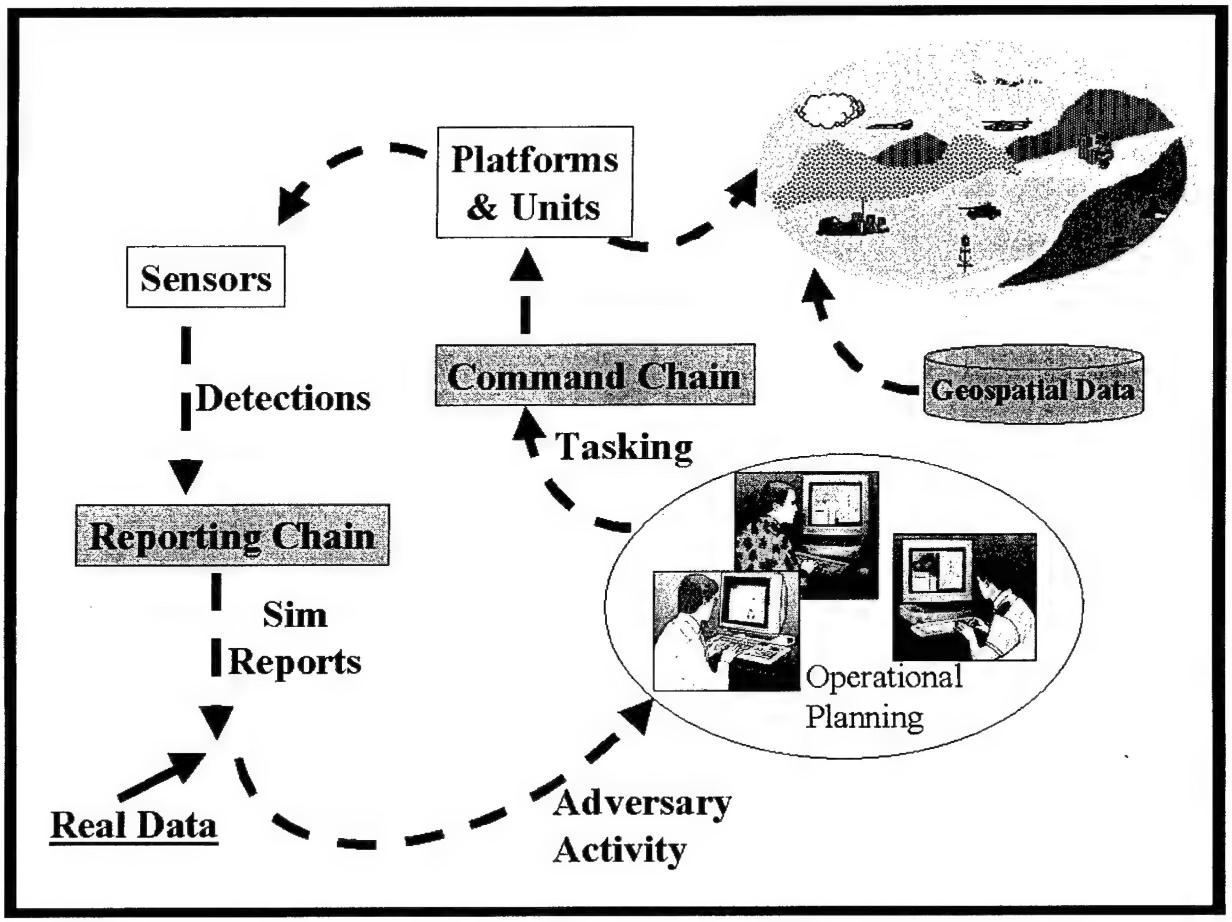


**Figure 2 Pictorial representation of required federation**

# 5. Demonstrable Federation

The format of this section is aligned with Steps 3 to 6 of the FEDEP.

## 5.1 Federation design

Many issues concerning running HLA on the EXC3ITE network could be adequately explored through use of an HLA-compliant version of DICE within a federation that only included itself or replications of itself. Exploring such issues as FOM development and code reuse among federates

required other federates within the federation. As such the goal federation and its federates are shown in Figure 3.

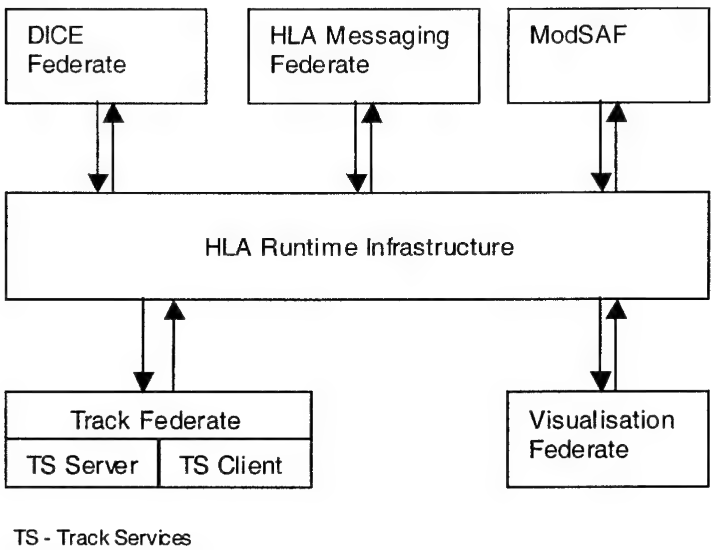


**Figure 3 HLA/EXC3ITE federation**

This federation would demonstrate

- Code reuse and the use of a C2 object model through the HLA Messaging federate and the DICE federate.
- Use of an internationally defined object model (RPR-FOM) through ModSAF, a visualisation federate and the Track federate.
- Use of an object model closely aligned to the Track Services IDL through the Track federate and the DICE federate.

Due to ModSAF and the visualisation federate using a previous version of the RPR-FOM these federates could not be utilised through HLA. Hence the federates able to interoperate through HLA were:

- DICE

*The DICE simulation software suite[10] would explicitly represent the reporting and command chains of the federation. Through appropriate interfacing, DICE would also enable the immersion of the CSS employed by the planners.*

* An HLA Track Federate

  *The client would enable a combination of simulated and real track data, managed by the EXC3ITE Track Services, to be used transparently within the HLA federation.*

* An HLA Messaging Federate

  *Given the limitation in suitable and available HLA-compliant products, this simple federate would permit demonstration of the reuse of object models and software libraries within the federation.*

Each of the above needed to be made HLA-enabled and compliant within the federation; they are each discussed in a dedicated section under Federation Development (Section 5.2). In addition the following products would be used to demonstrate overall capability:

* ADSIM

  *ADSIM[11] would model the required ground-based air defence radars and the motion, surveillance and engagement characteristics of maritime and airborne platforms. An existing ability to interoperate with the DICE simulation could be employed[4]. ADSIM would therefore provide ground truth of the sensors and the tasked air platforms.*

* DICE Track Services

  *These services (Appendix A) would provide the means by which simulated track data would be presented to the HLA Track Federate.*

* Live Track Services

  *Live track services such as those of TDRAP would provide the means by which real track data would be presented to the HLA Track Federate.*

* CSS for audience

  *The human audience of the resultant synthetic environment would employ the DSTO prototype Air Asset Visualisation Tool (AVT), the operational ADF Phoenix Display System (PDS) and messaging facilities provided by the DICE simulation software suite and the HLA messaging federate. The AVT displays temporal aspects of an air operation, in particular alert states and tasking. The PDS provides an air picture. The wargamers are therefore using a mix of operational and prototype systems. It is important that such systems do not necessarily show perfect information and that the wargaming planners work in an environment cognisant of delays and processes associated with Command, Control, Communication and Intelligence. This was achieved through using the DICE simulation with its existing ability to immerse the AVT and the PDS within a synthetic environment[4]. Synthetic track data is important for the stimulation and population of the PDS but rather than invent a different standard for dealing with simulated tracks, synergies are exploited between HLA and EXC3ITE by reusing the EXC3ITE Track Services as a stimulation service through HLA for the wargamers' systems.*

- ModSAF

  *ModSAF would model Land platforms. An existing ability to exchange data with the DICE simulation via a DIS interface could be employed.*

- MetaVR

  *MetaVR offered a means of visualising COA activities within a 3D environment. MetaVR has both a DIS and HLA capability.*

To take benefits from help desks and a large user community, it was decided to employ the DMSO RTI.

## 5.2 Federation development

The development of the three HLA federates listed in Section 5.1 is detailed in Appendix C and outlined in the following sections.

### 5.2.1 DICE

Within the DICE architecture, interfaces to other simulations and CSS are encapsulated as Peripheral Unit Interfaces (PUI) that can be configured and reused as required within any simulated scenario. PUI are based on a Common Node Framework (CNF) that provides a template and associated software libraries for any new PUI developments. One PUI is the Relational Database Management System (RDBMS) PUI that populates databases with simulated data, according to the simulated scenario, in a form accessible by the DICE Track Services (see Appendix A). DICE HLA developments concerned the use of the CNF to create a reusable HLA PUI.

The DICE simulation uses a standard formatted textual messaging language for internal communication[10]; interfacing to other simulations and CSS involves the use of data-driven mappings to and from the internal language. The DICE HLA developments were able to exploit this such that the simulation is able to employ any FOM without the need for software engineering. HLA interactions and messages will therefore be presented in a recognisable form to internal models of DICE that can then react to such inputs in a manner that their behaviour dictates.

For the purpose of the goal federation, it was possible to reuse existing DICE agents and interfaces to ADSIM, the AVT and PDS.

The DICE SOM (see Appendix C) describes the intrinsic C2 capabilities of the simulation which manifest primarily as the transmission and reception of formatted messages (DICEFORM) by C3I system 'nodes' represented by HLA *interactions*. The choice of interactions rather than HLA objects is based on the fact that interactions do not have persistance from an RTI standpoint. This decision is supported by findings in reference 13. By using the DICE SOM plus software libraries provided by DICE, other federates can choose to receive and utilise a complete DICEFORM or delve into the

structure (sets and fields) of the message content and retrieve specific information. Similar flexibility exists in the sending of DICEFORM. The SOM facilitates interoperation through HLA with military CSS that already have the ability to handle military formatted messages.

This particular DICE SOM provides a generic representation of messages which is powerful but it is acknowledged that supporting information regarding specific messages types, their structure and content, will be needed in order to use this. Also, details are needed of particular C3I system agencies being modelled within a DICE scenario and what they can provide other federates. A web-based environment is under development that provides linkages between the DICE SOM, a library of message interrogation and construction software, the DICE Message Definition Database (DMDD) and DICE scenarios.

Other HLA interactions needed to be designed and developed that enabled the DICE simulation control mechanisms of initialise, start, pause, resume and stop to propagate through to and be dealt with appropriately by other HLA federates.

The HLA compliance checks are outlined in Appendix B; the level of compliance achieved in this project is discussed in Appendix C. More development is needed in the areas of time management and dynamic object ownership.

### 5.2.2 An HLA Track Federate

The HLA Track Federate needed to be a standard client of EXC3ITE Track Services (see Appendix A) plus be HLA compliant with the simulation federation. The Track Service client was developed (see Appendix C) in accordance with two SOM, namely:

- An *IDL-SOM*. That is an object model aligned as closely as possible to the Track Service CORBA IDL.
- The Real-time Platform-level Reference FOM (RPR-FOM) that is an internationally used object model closely aligned with DIS protocols.

It is important to note that since the federate is compliant with the Track Service IDL, it can act as a client for any EXC3ITE Track Services and not just those of DICE. Thus the complete federate consists of the HLA Track Federate and whichever EXC3ITE Track Services the HLA Track Federate is a client of.

### 5.2.3 An HLA Messaging Federate

The HLA messaging federate is a minimal federate that enables the reception (subscription) and sending (publication) of DICEFORM through the utilisation of the DICE SOM libraries. DICEFORM can be composed through the user interface either by their full string representation or by setting the values of individual sets and fields. The DICE HLA PUI receives these messages and forwards them onto particular nodes within the DICE simulation. Which nodes receive these messages is configured through DICE. The user interface of the federate is shown in Figure 4.

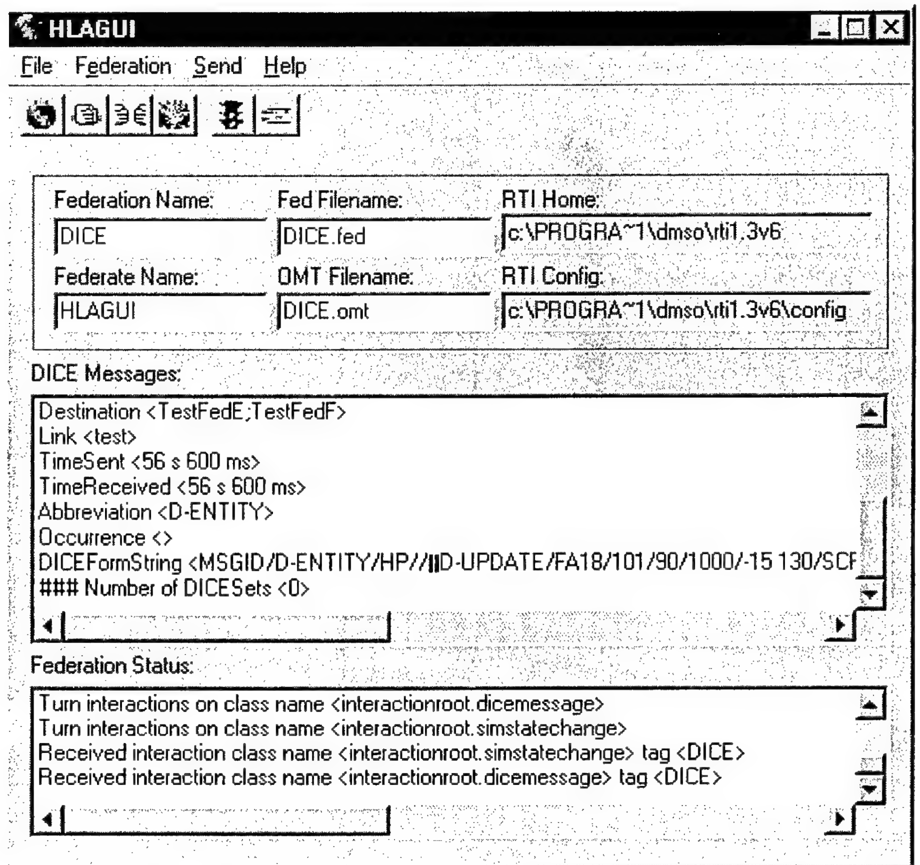


**Figure 4 User interface of HLA messaging federate**

### 5.2.4 Federation object model

The resultant FOM is discussed in Appendix C and is a combination of the DICE and IDL SOM and the RPR-FOM. No resolution of conflict was required in building the FOM from individual SOM.

## 5.3 Integration and testing

Once all federates had been developed it was necessary to integrate the federates into a federation and test data flows within the federation as shown in Figure 5.

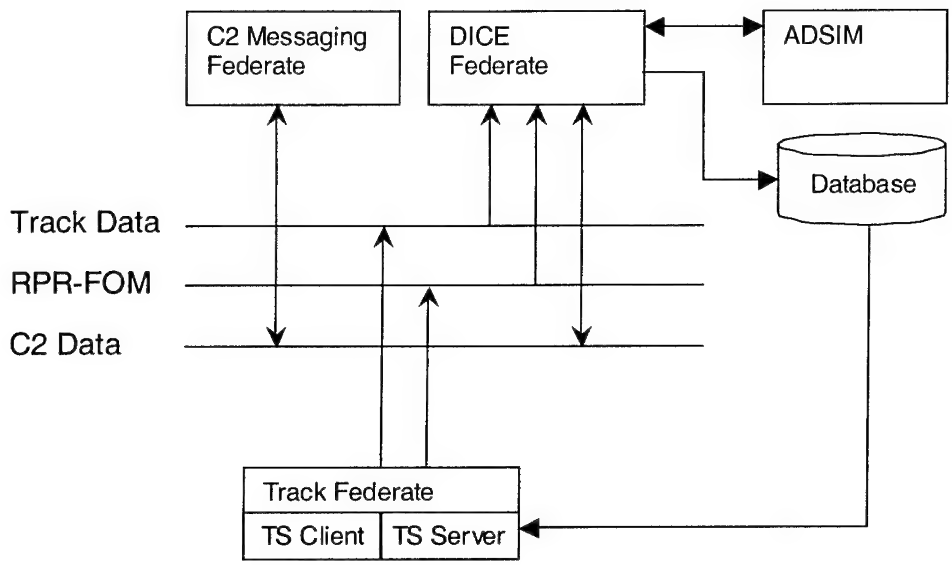


**Figure 5 Data exchange within the HLA/EXC3ITE federation**

The Track Federate can be used with any EXC3ITE Track Service; however, only the DICE Track Server was available for use in testing. The DICE Track Server utilises a database that during the simulation was populated by the DICE RDBMS PUI that in turn was fed data from ADSIM through the DICE ADSIM PUI. Ordinarily the database is populated by other means thus eliminating the need for the DICE federate to populate the database.

Figure 6 shows the configuration of overall synthetic environment that includes the HLA/EXC3ITE federation. An indication of the major information and command flows given earlier in the conceptual model is also shown in Figure 6. A walkthrough of the major flows and activities is given in Section 5.4.

In developing the HLA Track Federate, a conflict was discovered between RTI 1.3NGV2 and the CORBA ORBIX 3.0. This is discussed further in Appendix C.

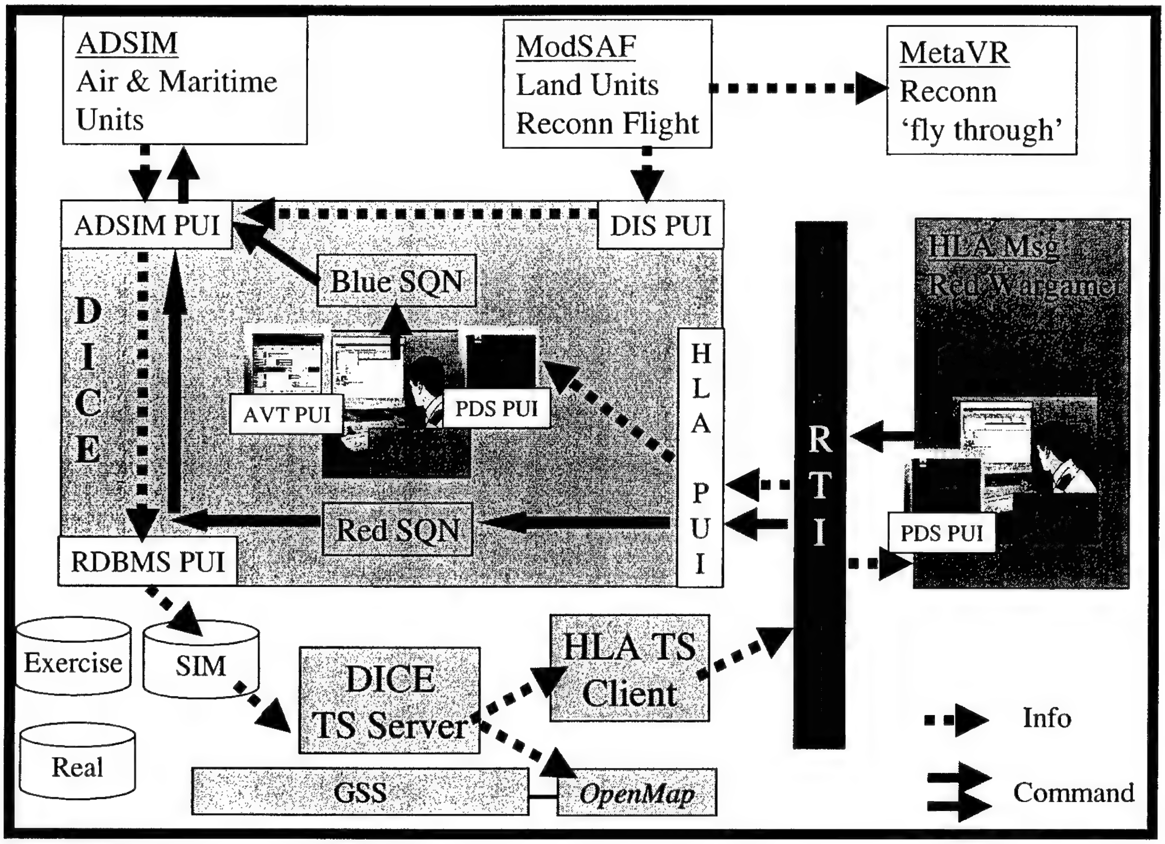


**Figure 6 Configuration of federation with major information and command flows**

## 5.4 Execution and analysis

The project culminated with a demonstration to the Takari Executive on 29 June 2000 that illustrated concepts and capabilities for building a modular distributed synthetic environment. The synthetic environment was geographically distributed between Salisbury, SA and Canberra, ACT and used an overarching theme of *Situation Awareness & Planning* to provide context.

### 5.4.1 Situation awareness

Military watchkeepers use a range of applications to monitor the status of own and adversarial forces and to assess the history, status and intent of the adversary. Such monitoring and assessment is aided by knowledge of environment, assets and asset movement. The first part of the demonstration illustrated the potential of EXC3ITE Track Services to contribute to information systems that aid situation awareness. Real-world data feeds were not available, so instead the DICE Track Service was used to access and query a database of notionally real data which was displayed and interrogated using the open-source software *OpenMap*[28] (see Appendix A) and Autometrics' Battlescape/NT. Air, land and maritime tracks were demonstrated. OpenMap was also used to access a range of geodata (eg map data and major towns and rivers) residing on different data servers distributed between Canberra and Salisbury. Figure 7 shows an illustrative OpenMap display.

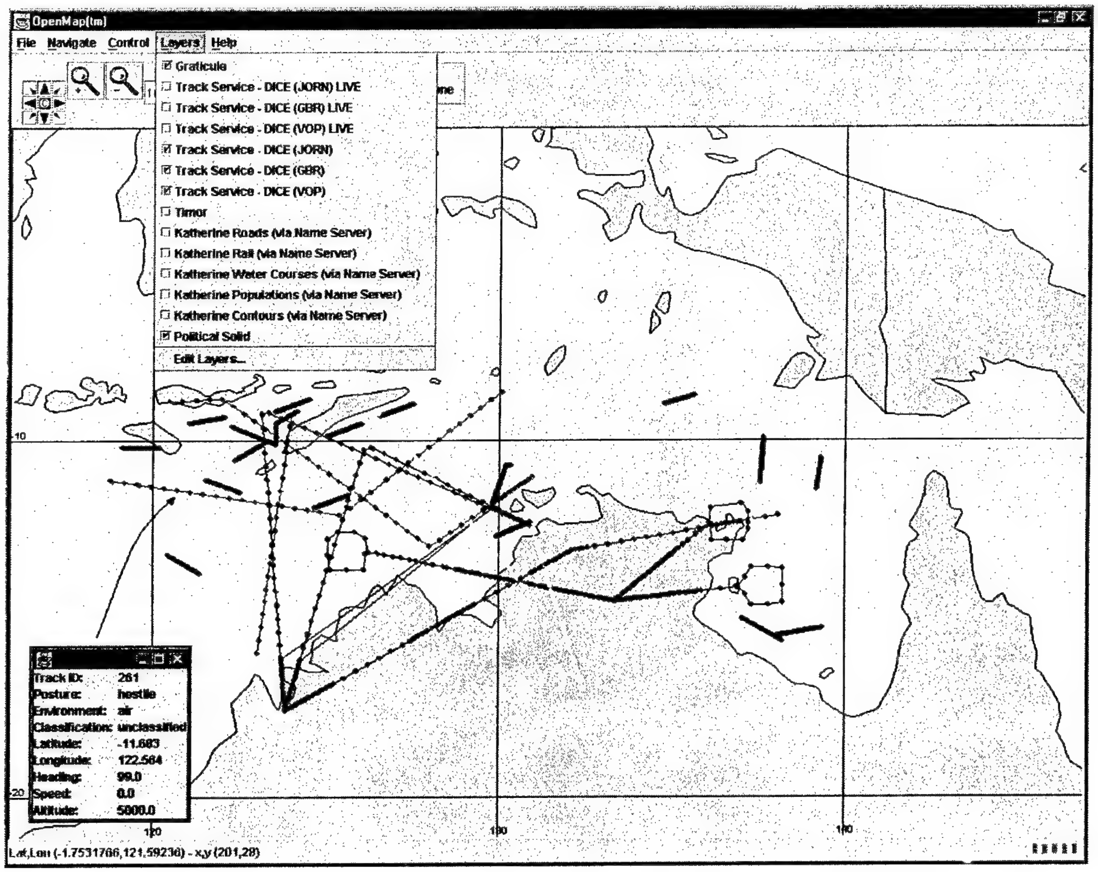


**Figure 7 OpenMap displaying EXC3ITE Track Service data**

## 5.4.2 Operational planning

The demonstration postulated the need for planning of defensive air operations and air reconnaisance flights. The distributed synthetic environment was used to enable 'Red' ('Kamarian') and 'Blue' wargamers to interact through the specification of friendly and adversarial COA and make run-time influences on COA execution. The qualities and behaviour of the synthetic environment are conveyed below using major flows and activities plus supporting comments. (It is important to stress that a spectrum of modelling and simulation tools is needed to support operational planning; synthetic environments are but one important element of this spectrum.) The synthetic environment could be initialised, started, paused, resumed and stopped according to control mechanisms employed within the DICE simulation controller facilities and propagated to other federates.

The synthetic environment was running at twice real time with federates and elements distributed between Canberra and Salisbury, SA as shown in Table 2:

| Canberra ACT | Salisbury SA |
|---|---|
| ADSIM and PUI | DICE Controller |
| Blue PDS and PUI | RTI |
| Blue DICE Human Player | HLA Track Federate |
| AVT and PUI | DICE Track Server |
| ModSAF | Red Msg GUI |
| Meta-VR VRSG | Red PDS and PUI |
| DICE DIS PUI | DICE RDBMS PUI |
| BattleScapeNT | DICE HLA-PUI |
| OpenMap | DICE Agents |

**Table 2: Synthetic environment distribution**

(i) Planning defensive air operations

- Both Red and Blue wargamers defined an initial posture and tasking.
  - The Red wargamer defined a COA consisting of a package of strike aircraft supported by fighters that gathered from Kamarian air bases and formed up for some mission over territory of strategic interest to Australia.
  - The Blue wargamer, based on current and expected intelligence, positioned an airborne early warning and control (AEW&C) aircraft plus fighters on combat air patrol (CAP).
  - The initial posture was reflected in ADSIM which represented the movement and physical characteristics of the Red and Blue air and maritime assets.
- Both Red and Blue wargamers' systems are stimulated by the synthetic environment (Figure 6):
  - Detection by Red and Blue surveillance assets of airborne platforms, plus position reports, are provided by ADSIM through the DICE/ADSIM PUI.
  - After required delays and filtering, the detections and position reports are output from DICE via the RDBMS PUI into the DICE Track Service relational databases of time-stamped simulated track data.
  - The DICE Track Service Server makes the DICE track service database information available according to the EXC3ITE Track Services IDL standard based on the demands of the HLA Track Federate.
  - The corresponding HLA objects are published by the HLA Track Federate via the RTI and using both the IDL SOM and RPR-FOM object models.

- The DICE HLA PUI, subscribing to the IDL SOM and RPR-FOM picks up the delayed and filtered detections and position reports and populates the Red and Blue PDS accordingly. Both the Red and Blue wargamers use their PDS to view the execution of COA.

- The synthetic environment is able to respond to interactions made by the wargamers (Figure 6):

  - Based on information presented in the PDS, the Blue war gamer used the DICE messaging facility to create an order tasking CAP aircraft to intercept the Red package. The order is passed to the Blue Squadron, a behavioural Petri net simulation in DICE, which, after associated delays and processes, influences accordingly via the DICE/ADSIM PUI the aircraft modelled in ADSIM.

  - The Blue wargamer scrambled additional fighters from Darwin to replenish the CAP by sending orders to the Blue Squadron. After associated C3 delays, the Squadron produces updates the animated Gantt schedule of the AVT and also passes scramble directives to the ADSIM simulation which is modelling those aircraft.

  - Via the stimulation mechanisms discussed earlier, the wargamer directives result in subsequent updates to the air picture displayed in the PDS systems which were used to view the closing stages of the COA. The interception of Red aircraft by Blue, corresponding detections and engagements were modelled in the ADSIM simulation and the outcomes relayed to the planners from ADSIM. In the case of the Red wargamer, the outcomes are reported in the form of messages received by the HLA Messaging Federate from the DICE HLA PUI using the DICE C2 SOM. As a consequence, the Red wargamer used the HLA Messaging Federate to create a message ordering all remaining Red aircraft to return to base. The message was communicated using the DICE C2 SOM via the HLA PUI, the behavioural Red Squadron Petri net agent in DICE and the ADSIM PUI.

- The Red and Blue tracks could be viewed in real time using the OpenMap client of the EXC3ITE Track Services.

(ii) <u>Planning air reconnaisance</u>

- The ModSAF simulation was populated with own and adversarial ground units in the region of interest. The Blue planner used ModSAF to specify the route of and model air reconnaisance flights.

- ModSAF provides DIS protocol data units to (Figure 6):

  - the DICE DIS PUI resulting in population of ADSIM with the reconnaisance flights; and

- MetaVR VRSG, enabling visual fly-throughs and inspection of terrain and terrain features, the likely positions of enemy units and the possible flight paths and tasks of the reconnaisance flights.

- It was also possible to run a pre-recorded reconn flight out of the EXC3ITE Track Services:

  - The HLA Track Federate conveys positional information using the RPR-FOM which is communicated to the DICE HLA PUI via the RTI;

  - Once internal to DICE, the information is passed to the DICE DIS PUI and subsequently to MetaVR using the DIS protocol.

### 5.4.3 Achievement of aims

The distributed demonstration achieved the demonstration objectives discussed earlier; namely:

- Demonstrate a hybrid of M&S federates and EXC3ITE federates;

  *EXC3ITE Track Services designed for use with real data within real C3I systems (as conveyed by Situation Awareness segment of demonstration) were able to interoperate with HLA federates through use of the HLA Track Federate acting as a Track Service client and an HLA federate.*

- Demonstrate a level of synergy and reuse between HLA and EXC3ITE;

  *Reuse was demonstrated of the efforts of and products developed by the EXC3ITE Track Service Focus Group. The HLA Track Federate communicated using a SOM that mirrored the EXC3ITE Track Service IDL. DICE was able to interoperate with the HLA Track Federate using this SOM in the air operations planning phase.*

- Demonstrate a presence of C2 simulation with HLA;

  *A powerful HLA-enabled version of DICE was developed that has extensive capabilities. A DICE C2 SOM was designed and demonstrated that addresses the use of formatted textual messages to convey a broad range of orders, reports and requests not possible with DIS.*

- Demonstrate reuse of own SOM and software libraries;

  *The DICE HLA software libraries are designed to facilitate use by other federates. The HLA messaging federate was able to make use of these libraries to communicate through the DICE C2 SOM in the air operations planning phase.*

- Demonstrate an ability to adopt an existing international SOM;

  *The HLA Track Federate communicated using the RPR-FOM; interoperation with DICE was demonstrated using this object model during the air reconnaisance phase.*

- Demonstrate a hybrid of HLA and other 'standards'.

  *A range of simulations were able to interoperate using a combination of HLA, DIS, the DICEFORM standard, and DICE interfaces perculiar to ADSIM, PDS and AVT to form a distributed synthetic environment able to immerse wargamer systems. The air reconnaisance planning segment demonstrated a flow of information that originated from the EXC3ITE Track Services using the Track Service IDL to be displayed in MetaVR VRSG via both HLA and DIS.*

# 6. Discussion

## 6.1 EXC3ITE-based simulation and synthetic environments

Executable models and simulations include those of sensors, weapon systems, platforms, C3I system agencies such as a headquarters, and communications. Models and simulations need to be wrapped and described in some open and standard form in order to facilitate and maximise their interoperability and reuse so that particular federates can be more easily built for diverse purposes. Federations need to be able to be built quickly as required from appropriately packaged reusable components or building blocks. The HLA/EXC3ITE work program described in this document concerned the extent to which the adoption of HLA in conjunction with and as a standard within the EXC3ITE environment can help achieve this. The project illustrated that the common architectural principles shared by HLA and EXC3ITE can aid interoperability and reuse between the architecture and practices of real C3I systems and that of simulation and synthetic environments. It is important to note that HLA alone is not a panacea. In scoping and pursuing the role and desired capabilities for EXC3ITE-based simulation and synthetic environments it is critical to recall the context within which the HLA concept was initiated, as discussed in Section 3. HLA can only be successfully adopted in conjunction with pursuing the other initiatives of the US M&S *common technical framework*.

Data standards and conceptual models that convey a common view are comparable with the goals of the EXC3ITE Track Services and GSS focus groups, and EXC3ITE ideals in general. In the case of EXC3ITE Track Services, the project showed that common data standards and conceptual models can be identified between simulation and real systems. Suitable products were not readily available for exploring and establishing a strong relationship between the evolving GSS and simulation. This issue needs to be addressed; in particular, the relationship required between GSS and the SEDRIS standard and tools.

Supporting common software and tools, and resource repositories is also of significant importance.

The following view is given on EXC3ITE simulation and synthetic environment *services* which builds on that reported in reference 14.

### 6.1.1 Model and simulation repositories

Model and simulation library services should be made established that make available:

- Model description

  For example, if wishing to experiment with different C2 structures and processes using a number of optional simulations, HLA object models help by conveying the input/output characteristics of a simulation but need to be complemented with accessible descriptions of intrinsic behaviour.

- Model history and ownership;
- Level of fidelity and resolution;
- Validation, verification and accreditation;
- Hardware/software requirements;
- HLA Simulation/Federation Object Models and software to interrogate and use them; and
- Inputs, outputs and their formats and medium.

### 6.1.2 Stimulation services

Stimulation refers to the ability for models and simulations to drive other systems. Within the EXC3ITE environment such systems might include operational CSS plus a range of experimental or prototype technologies and systems. Pure stimulation refers to the generation and injection of stimuli but with no ability for the stimulator to respond to any behaviour that occurs within the system being stimulated. Within the EXC3ITE environment, synthetic stimulation could complement feeds from real-world sources, albeit a challenge. Pure stimulation is relatively easy to achieve compared with immersive capabilities described later.

Models and simulations may be executed and their outputs packaged as run-time stimuli for some other system in a true 'push' fashion. Alternatively, or in addition, such models and simulations could be executed and their outputs wrapped and stored for later use. Such stimulation data could be static or time-stamped to enable scheduling. Hence stimulation services could access data created by 'off-line' model execution in a 'pull' manner but achieving a pseudo 'push' through scheduling. An example of a stimulation service is that of the DICE Track Services (Appendix A).

A key aspect of stimulation is to manage levels of 'seamlessness'; ie the degree to which the stimulated system is driven in a manner equivalent in look and feel to its real-world stimuli. This necessitates the management of simulated versus real data.

### 6.1.3 Synthetic environments

Synthetic environments refer to the ability for models and simulations to not only stimulate but immerse and respond to behaviour within any system under study, training or testing[15]. Such behaviour will generally manifest as orders, reports and requests that need to be accommodated by entities or agencies within the synthetic environment. This includes the concept of playing out a plan within a synthetic environment for the purposes of assessment, rehearsal or training. Controllable and composable synthetic environments are essential to the demonstration, trialing and assessment of emerging or prototype technologies and systems that require interactive experimentation. Of particular importance to the C2 enabling purpose of EXC3ITE is that synthetic environments are C2-led in that C2 is represented explicitly whilst permitting consideration of environment and context or scenario. Within EXC3ITE, synthetic environments could be melded with real-world or live components. The HLA/EXC3ITE project demonstrated concepts and capabilities of an immersive synthetic environment and built on earlier capabilities reported in references 4 and 16. There are particular challenges associated with the development of 'services' in this area.

Whilst the realisation of a 'desktop simulation service icon' that anyone could click on and run over EXC3ITE could occur more readily for stimulation services (specifically post-execution), the issues concerning achieving this in the case of synthetic environments are significant and should not be underestimated. Aside from managerial (time and simulation/scenario), technical and behavioural modelling matters, there are considerable issues related to the need for manual interaction. Many models need to be manually initiated and executed; simulation-based wargaming comes at a price of manually intensive human response cells that act as the interface between the training or subject audience and the synthetic environment. Again, HLA will help in many areas but it is not a panacea.

### 6.1.4 Common software tools

These may include:

- Various HLA RTI;
- Software enabling the parsing and creation of numerous protocols such as HLA SOM; DIS Protocol Data Units (PDU); and real C3I system messages such as ADFORMS;
- After-Action Review and other analytical tools

### 6.1.5 Supporting services

A range of services applicable to real systems need to also be able to support simulation; these include:

- Geospatial 'data' and 'process' services (eg GSS);
- Track services;

- Translator Services: whilst their proliferation should be avoided, in the absence of global standards there will always be a need for bridges and translators that enable disparate systems to interoperate. Capabilities here include DIS to HLA gateways and message translation utilities such as ADFORMS to OTH-T Gold. Whilst the DICE simulation can employ military messages within the simulation (in the interest of semantic and syntactic equivalence and interoperation with real C3I systems), some translation is still required when interoperating with other models and simulations. Translator services may include one-to-one through to many-to-many configurable 'gateway' capabilities.

## 6.2 HLA and C2

The Northern Atlantic Treaty Organization (NATO) counties recently published a code of best practice for modelling and analysing C2[12]. The publication included guidelines that stressed the need for 'C2-based' modelling to be able to represent:

(i) Information as a commodity (ie a resource that can be collected, processed and disseminated and has dynamic attributes concerning accuracy, relevance, timeliness, completeness and precision);

(ii) Realistic information flow around the battlespace (ie its source, information loss and degradation);

(iii) Collection of information from multiple sources and the tasking of information collection assets;

(iv) Processing of information (ie any filtering, correlation, fusion etc from its original form);

(v) C2 systems as entities on the battlefield;

(vi) Individual 'unit' perceptions built, updated and validated from the information available to that unit;

(vii) Commander's decision based on perception of the battlespace; and

(viii) Information Operations (ie represent deliberate attack and protection of information, information systems and decisions).

Implicit in the above is the need to represent C2 structures, functions, processes and systems, and command decision-making. Whilst satisfying the NATO criteria requires achievement of certain behavioural representation properties within models and simulations, it also necessitates appropriate simulation architecture and architectural practices. The DICE simulation software suite was purposely designed to facilitate this and much of the above can be readily achieved within the DICE environment. The DICE HLA development is a step towards greater standardisation and closer synergies with international developments and future real C3I systems. HLA has a key role in achieving C2-led simulations and synthetic environments. With reference to

the NATO criteria, the following key points are made concerning the role of HLA; many issues need further attention:

- *Information as a commodity; its collection, flow and processing*

  The DICE HLA development enable *messages* that concern a range of orders, reports and requests to be modelled explicitly and in a manner that is HLA-compliant. It enables the available outputs of a C2 simulation to be packaged and presented for consideration and use by other federates. The collection, flow and processing of such messages can again be modelled explicitly in conjunction with appropriate simulations. The messages enable the tasking of collection assets. Manipulating and controlling dynamic attributes of information, such as degradation, is evidently a significant management issue and implications on HLA object models, objects and interactions needs to be explored.

- *C2 systems as battlefield entities; information operations*

  C2 systems and agencies need to be able to be targeted, degraded and destroyed by physical or non-physical means. Appropriate standardisation leading to design of corresponding HLA interactions is needed for addressing non-physical influences. The ability to use and complement the RPR-FOM (comparable with the DIS PDU) to convey the location, physical behaviour and damage etc of C3I systems and agencies needs to be investigated.

- *Language and format*

  It is felt that HLA object models and interactions based on formatted textual messages should adequately address the C2 language issues. In reference 17, DMSO appeared reluctant to use formatted messages for C2 simulation but rather pursue a solution based on 'extracting the primitives' of a message and incorporating them into the FOM. The DICE SOM and software libraries (Appendix C) enable a federate to publish and make use of a whole message or specific fields or atomic elements. The SOM is based on the DICEFORM language which mirrors that of real military messaging and can incorporate a range of military standards such as ADFORMS, OTH-T Gold and ADGE SITS. The language is designed to be both human and machine-readable. Importantly, custom messages can be created. Messages can be complex, employing repeated fields and sets and elements of free text, or simplistic (2 or 3 structured fields). This flexibility is important to C2 and should provide a means of conveying matters such as commander's intent and perceptions in addition to orders, reports and requests. Formatted textual messages can be used to convey to simulations information that is voice-told in the real world.

  The US Command and Control Simulation Interface Language (CCSIL) was designed to permit C2 messages to be passed within DIS Signal PDU. Reference 17 confirmed opinions that the design rationale for CCSIL was flawed and that the project is not being further funded. Any Australian reliance on CCSIL would therefore be a bad investment.

- *Object model standardisation*

  Object model standardisation is needed that maximises alignment with real C3I system developments. This is an important issue that faces realisation of a 'plug and play' capability using HLA. Standardisation applies to the messages used to convey C2 matters as well as the elements found within a C2 message. The DICEFORM language and hence DICE SOM enables a range of message field templates or formats to be employed (eg different formats for conveying positional information). In reference 17, DMSO discussed an aspiration for a common information model within the M&S community including evolution of a Common Object Definition Dictionary, enabling reuse of data atoms. Any standards need to be managed appropriately.

- *Interfacing and interoperability of simulations with real C3I systems?*

  Capabilities that enable real and prototype C3I systems to interoperate with simulated systems are considered to be critical by the international simulation community[17,18]. In fact, the need for such capabilities is a key reason for maximising a relationship between EXC3ITE, simulations and synthetic environments. The US DMSO is pursuing HLA as the medium by which real systems and simulation interoperability is explored and achieved. Reference 17 reported that the US considered it necessary to bring HLA software components into the US Defense Information Infrastructure Common Operating Enviroment (DII-COE). The goals here rely on common standards, data and object models between real and simulated systems; such commonality is in its infancy in Australia but EXC3ITE provides an ideal capability for pursuing this. A phased approach is needed starting with message-based approaches through to common object models, databases and infrastructure resulting in a seamless presence of HLA within operational C3I systems themselves.

A focus area under The Technical Cooperation Program (TTCP) Joint Systems Analysis (JSA) Group Technical Panel 2 (Modelling and Simulation) has been established to explore "Simulation-C4ISR Interoperability".

## 6.3 Other issues

The conflict between RTI 1.3NGV2 and the CORBA ORBIX 3.0 does not enable the use of naming servers within an HLA federate. Naming servers have a key role in ensuring transparent connectivity within EXC3ITE and hence the significance of this issue will need to be explored further.

Whilst the distributed nature of DICE worked well between Salisbury and Canberra, difficulties prevented distribution of the HLA Track Federate and RTI on different EXC3ITE subnets. This needs to be investigated further.

# 7. Conclusions

The project was a pathfinder for EXC3ITE-based simulation and synthetic environments and provides a foundation for further R&D. The project illustrated that interoperability and reuse are achievable between the architecture and practices aspired to for real C3I systems and that of simulation and synthetic environments. Experiences from the project can inform projects such as the Virtual Air Environment and the evolution of the Australian Defence Simulation Office.

The existence of a standard language within DICE permits its participation in a range of HLA exercises with minimal software coding. The deliverables from this project form important elements of ITD's Joint Synthetic Environment Facility (JOSEF) and aid exploration of the synthetic environment element of an integrated modelling environment for operational planning.

This report describes the capabilities developed and experiences gained and raises issues and recommendations on the way ahead for EXC3ITE-based simulation and synthetic environments. The project demonstrated use of HLA to convey orders, reports and requests, or *C2 messages*, modelled explicitly by the DICE simulation. Investigation is needed into the means by which HLA can enable C3I systems and agencies to become battlefield entities that can be subjected to targeting, degradation and destruction by physical or non-physical means. It is felt that HLA object models and interactions based on formatted textual messages should adequately address C2 language issues and that the DICE HLA development offers a significant capability that will aid interoperation with real C3I systems. C2 object model standardisation is needed that maximises alignment with real C3I system developments and works towards a 'plug and play' capability using HLA. Capabilities that enable real and prototype C3I systems to interoperate with simulated systems are critical and this is a key reason for maximising a relationship between EXC3ITE, simulations and synthetic environments.

The project has established a foundation upon which such issues can be explored. A TTCP focus area on "Simulation-C4ISR Interoperability" is intended to further explore the significance of HLA to C2.

The baseline EXC3ITE simulation capability provided by this project needs to be maintained and extended.

# Acknowledgement

The authors would like to thank staff of Aspect Computing and Adacel Technologies for the development and delivery of the HLA/EXC3ITE products and acknowledge the input of Dr Darryn Reid into the scoping of the project.

# References

1. http://exc3ite/

2. http://www-itd.dsto.defence.gov.au/IMF/TrackServices/index.shtml

3. DSTO EXC3ITE Track Service Focus Group. *'EXC3ITE Track Service Specification for CORBA V1.1'*, December 15 1999. Draft DSTO document.

4. Matthews,K., Davies,M., Dunn,J.M., Gabrisch,C., *'Synthetic Environments for C3I Experimentation'*, Proc. SimTecT2000 conference, Sydney, Australia, 28 February – 2 March 2000.

5. http://www-itd.dsto.defence.gov.au/IMF/GeoService/index.shtml

6. *The Role of DMSO'*, Military Training & Simulation News, Vol.2, Issue 3, June 2000, pp14-16

7. http://www.dmso.mil

8. http://hla.dmso.mil/standards/

9. http://www.sedris.org/

10. Davies,M., Gabrisch,C., Dunn,J.M., Bowden,F.D.J. "The Distributed Interactive C3I Effectiveness (DICE) Simulation". DSTO Technical Report, DSTO-TR-0485, February 1997

11. Haydon,N.A., "Establishing An Interface To A DOS-Based Air-Picture Simulation Via A Local Area Network", DSTO General Document, DSTO-GD-0117, November 1996

12. Northern Atlantic Treaty Organization (NATO) Research and Technology Organization (RTO), "Code of Best Practice (COBP) on the Assessment of C2", RTO-TR-9 AC/323(SAS)TP/4, March 1999

13. Carr,F.H., Roberts,J.D., "Incorporating Command and Control into the High Level Architecture (HLA): An Architecture for Realistic Communication Effect", Proc. Simulation Interoperability Workshop 8-12 September 1997, 97F-SIW-076

14. Vaughan,J., Davies,M., 'EXC3ITE Simulation Service: bringing simulation into the operational domain', Proc. SimTecT2000 conference, Sydney, Australia, 28 February – 2 March 2000, pp39-44

15. Matthews,K., Bourn,S., Davies,M., 'Simulation and Synthetic Environments for Command and Control of Air Operations', DSTO Technical Report, DSTO-TR-1096, January 2001

16. Grisogono,A.M., Vaughan,J., Menadue,W.I., Seymour,R.S., Davies,M., 'Synthetic Environments In Support Of Capability Development: Design Case Study For The Armed Reconnaissance Helicopter At Exercise Phoenix', Proc. SimTecT99 conference, Australia, 29 March – 1 April 1999, pp369-375

17. Davies,M., " Report on Visit to the UK and USA: 14 May to 1 June 1999 (Unclassified Aspects)', Overseas Visit Report, DSTO-OR-0315, August 1999

18. Davies,M., 'Report on visit to Paris, France: 10 to 17 January 1999', Overseas Visit Report, DSTO-OR-0282, January 1999

# Appendix A: DICE Track Services

Target tracking systems typically produce a series of track updates that are stored in databases and disseminated as messages that can have a range of disparate formats. Each track update (a *track point*) provides an updated estimate of a particular target that is being tracked. A sequence of track updates pertaining to the same target constitutes a *track* of that target over time.

As described in Section 2.1 the purpose of EXC3ITE Track Services is to provide CORBA compliant interfaces that are applicable for any repository of target track data. Such interfaces will enable a repository of track data to present itself within EXC3ITE as a track service while hiding the internal details of that repository. The EXC3ITE track service specification[3] is intended to be a guide for software engineers to implement the interfaces on the server-side and in user applications. The formal EXC3ITE track service specification uses the OMG Interface Definition Language (IDL).

## A1 The EXC3ITE track service specification and IDL

From samples of actual surveillance track data it is apparent that a range of different properties can be present in each track point and the set of properties may differ from one track point to the next. For example, a radio call sign or IFF code can appear intermittently throughout a track. Hence, the EXC3ITE track service specification identifies a standard set of track properties along with their value types to provide an agreed representation for sharing such data between track servers and client applications.

Track services provide track data to clients through the interaction of a number of interfaces. These interfaces enable a client to pull information from a track service. The track service IDL consists of the following interfaces and objects.

The `TrackCollectionFactory` interface is the entry-point for clients of a track service. It enables a client program to ask a server to create a single `TrackCollection` object that contains a potentially large number of `Track` objects.

The `TrackCollection` interface provides an efficient way of handling potentially many `Track` objects by exposing them through a single `TrackCollection` object. Collection objects are typically advertised in a naming or trader service. This interface allows `Track` objects to be accessed either as a sequence of references to those objects or through the `TrackIterator` interface.

The `TrackIterator` interface enables the client to control how many `Track` objects are returned with each access. The `TrackIterator` object maintains an internal pointer to the current `Track` inside the collection. There is no special ordering on the `Track` objects.

The `Track` interface exposes the mandatory properties of target tracks including their track points. The track points are ordered in time and there are potentially hundreds of them. This interface allows track point data to be accessed in several different ways that are suited to different

purposes. Those methods include `TrackPoint` objects being accessed either as a sequence of references to those objects or through the `TrackPointIterator` interface.

The `TrackPointIterator` interface enables the client to control how many `TrackPoint` objects are returned with each access. The `TrackPointIterator` object maintains an internal pointer to the current `TrackPoint` and is initialised to point at the earliest `TrackPoint`. `TrackPoint` objects are ordered in time.

The `TrackPoint` interface exposes the mandatory and optional properties of `TrackPoint` objects using several different methods.

Figure A1 shows excerpts from version 1.0 of the EXC3ITE Track IDL, including the interfaces described above (see [3] for the complete EXC3ITE Track Service IDL).

```
//-----------------------------------------------------------
// EXC3ITE TRACK SERVICE IDL, VERSION 1.0
//-----------------------------------------------------------

#include "OGIS.idl"
#include "Exc3iteTime.idl"
#include "Exc3iteSecurity.idl"

module Exc3iteTrack
{
    // Exc3iteTrack Specification, version 1.0.
    //
    // Produced by the Exc3ite Track Service Focus Group,
    // 2 September 1999.
    //
    // Refer to the specification notes accompanying this IDL.

    //-----------------------------------------------------------
    // Forward declarations of interfaces
    //-----------------------------------------------------------

    interface TrackCollectionFactory;
    interface TrackCollection;
    interface TrackIterator;
    interface Track;
    interface TrackPointIterator;
    interface TrackPoint;

    //-----------------------------------------------------------
    // Well-Known Properties and their types
    //-----------------------------------------------------------

    const long WKPTime = 1;                 // Exc3iteTime::Time
    const long WKPTimeStruct = 2;           //     TimeT
    const long WKPTime_hr = 3;              //     TimeInHoursT
    ...

    struct ErrorEllipse {
        double lenSemiMajor_nm;
        double lenSemiMinor_nm;
        double bearingMajor_degt;
    };

    enum OpEnv {sub_surface, surface, land, air, space};

    enum Posture {unknown, friend, neutral, hostile, suspect, faker, bogey};
```

```
typedef string <4> IFFMode;
struct IFF {
    IFFMode model;
    IFFMode mode2;
    IFFMode mode3;
    IFFMode mode4;
};

struct IVPair {
    long id;
    any value;
};
typedef sequence <IVPair> IVPairSeq;

struct Profile {
    sequence <OGIS::Envelope> area_list;
    boolean earliest_set;
    Exc3iteTime::TimeT earliest;
    boolean latest_set;
    Exc3iteTime::TimeT latest;
    sequence <string> track_id_list;
};

typedef sequence <Track> TrackSeq;
typedef sequence <TrackPoint> TrackPointSeq;

struct Point {
    Exc3iteTime::TimeT time;
    OGIS::WKSPoint position;
    OpEnv op_env;
    Posture posture;
    string track_id;
    Exc3iteSecurity::Label security;
};
typedef sequence <Point> PointSeq;

//----------------------------------------------------------
// TrackCollectionFactory interface
//----------------------------------------------------------

interface TrackCollectionFactory
{
    exception InvalidProfile {};

    TrackCollection get_tracks(in Profile user_profile)
        raises (InvalidProfile);
};

//----------------------------------------------------------
// TrackCollection interface
//----------------------------------------------------------

interface TrackCollection
{
    readonly attribute long number_tracks;

    TrackSeq get_track_sequence();
    TrackIterator create_track_iterator();
    void free();
};

//----------------------------------------------------------
// TrackIterator interface
//----------------------------------------------------------

interface TrackIterator
{
    exception IteratorInvalid {};
```

```
    boolean next(out Track track) raises (IteratorInvalid);
    boolean next_n(in short n, out TrackSeq tracks)
        raises (IteratorInvalid);
    void reset() raises (IteratorInvalid);
    void free();
};

//-----------------------------------------------------------
// Track interface
//-----------------------------------------------------------

interface Track
{
    exception InvalidProperty {};

    readonly attribute string track_id;
    readonly attribute Exc3iteSecurity::Label security;
    readonly attribute TrackPoint latest_point;
    readonly attribute long number_points;

    PointSeq get_point_sequence();
    TrackPointSeq get_trackpoint_sequence();
    TrackPointIterator create_trackpoint_iterator();
    string get_display_name(in long property_id)
        raises (InvalidProperty);
    void free();
};

//-----------------------------------------------------------
// TrackPointIterator interface
//-----------------------------------------------------------

interface TrackPointIterator
{
    exception IteratorInvalid {};

    boolean next(out TrackPoint point) raises (IteratorInvalid);
    boolean next_n(in short n, out TrackPointSeq points)
        raises (IteratorInvalid);
    boolean previous(out TrackPoint point)
        raises (IteratorInvalid);
    boolean previous_n(in short n, out TrackPointSeq points)
        raises (IteratorInvalid);
    void reset_to_earliest() raises (IteratorInvalid);
    void reset_to_latest() raises (IteratorInvalid);
    void free();
};

//-----------------------------------------------------------
// TrackPoint interface
//-----------------------------------------------------------

interface TrackPoint
{
    exception InvalidProperty {};
    exception InvalidConversion {};
    exception PropertyNotSet {};

    readonly attribute Exc3iteTime::Time time;
    readonly attribute OGIS::WKSPoint position;
    readonly attribute OpEnv op_env;
    readonly attribute Posture posture;
    readonly attribute string track_id;
    readonly attribute Exc3iteSecurity::Label security;
    readonly attribute Point point;

    IVPairSeq get_property_sequence();
    boolean property_exists(in long property_id)
```

```
            raises (InvalidProperty);
        any get_property(in long property_id)
            raises (InvalidProperty, PropertyNotSet);

        Exc3iteTime::Time get_time(in long property_id)
            raises (InvalidProperty, PropertyNotSet,
            InvalidConversion);
        OGIS::WKSPoint get_point(in long property_id)
            raises (InvalidProperty, PropertyNotSet,
            InvalidConversion);
        ...
        string get_display_name(in long property_id)
            raises (InvalidProperty);
        void free();
    };
};
```

**Figure A1: The EXC3ITE Track Service IDL Version 1.0.**

## A2 The DICE track service

There are currently three DSTO R&D systems that accommodate EXC3ITE track data, namely ITD's DICE (simulated track data), LOD's LSA-C4ISR (simulated track data) and SSD's TDRAP (real track data). The DICE track service will be discussed here.

The DICE simulation software suite[10] enables simulations and synthetic environments to be composed in a manner that is C2-led, in that C2 and intelligence processes plus communications and information flows are represented explicitly. This, plus the ability to employ real-world military messages and interface to operational command support systems (CSS), makes DICE a key tool for EXC3ITE. C2-led simulations also require some form of representation of the physical environment. The physical domain model contains representations of contesting force assets such as sensors, weapons, aircraft, ships and troops. The ADSIM[11] and ModSAF physical domain models are used in this research activity. The following paragraph describes some typical information flows during execution of a DICE simulation that is providing simulated data for EXC3ITE.

Throughout the simulation ADSIM (and/or ModSAF) provides sensor detections and asset reports to DICE that stimulate the C2 models. The C2 models respond to these detections and asset reports, process them in some way, and subsequently outputs messages to databases on servers in EXC3ITE. These messages may contain track information (track data available from DICE includes Jindalee Over-the-horizon Radar Network (JORN), Ground Based Radar (GBR) and Visual Observation Post (VOP)[4]) or intelligence information (from VOP or other). This track and intelligence data is available from the DICE simulation as a run-time or post-simulation service. Other messages (ie orders, reports, requests) may also be output during a simulation. These messages may be ADFORMS or OTH-T Gold messages that may be accessed by command support systems such as JCSS or BCSS or they could be simulated voice told messages. Track data is currently available to EXC3ITE via the DICE track service. Intelligence data and messages could be made available in a similar manner.

The simulated information flows within DICE are provided as formatted messages, either in standard forms such as ADFORMS and OTH-T Gold, or custom-designed messages. These messages are passed to a node or nodes that store these messages in relational database tables. These nodes, called relational database management system (RDBMS) peripheral unit interfaces (PUI), use a format specification to indicate the mapping of fields in the messages to columns in the database tables. These tables can then be interrogated by the DICE track service(s) to provide run-time or post-simulation data to clients. The DICE track service is implemented in C++ and uses ORBIX CORBA for communication with clients and naming services.

An OpenMap client has been developed to access EXC3ITE track service data. OpenMap is an open-source, freely available client produced by BBN technologies. OpenMap is a Java Bean based application that can display many different forms of map data and can handle zooming, different projections and mouse interactions. Since it is written in Java, OpenMap is portable and easy to understand, and its Bean base allows features to be used as components in other applications. Being open-source the code for OpenMap is freely available and can be modified to suit different needs. OpenMap displays its picture in layers, and comes with an ESRI shapefile depicting the world as one layer. This layer can be overlaid with other layers depicting, for example, roads, rivers, centres of population, contours and so on. Such a layer is being created to display track data.

In order to access the DICE track service and display it using OpenMap, the server and client are implemented according to the EXC3ITE track service IDL. This IDL is compiled to produce stubs for the server and client applications, such that the format of the inputs and outputs of servers and clients are common across implementations. The DICE track service uses the information from clients to structure queries on the databases using embedded SQL and returns the results of these queries to the client in the format mandated by the IDL. This achieves transparency across implementation of track services, as regardless of the underlying format of the data that the track service uses, the client will always receive response data in a known format irrespective of the track service instance.

The OpenMap client is modified to function as a layer class displayed by the OpenMap viewer application. The client layer class reads in the same information from the server via CORBA but uses it to display a track – a sequence of positions on the screen (points connected by a line). The sequence of events and transactions that occur between OpenMap, the CORBA Naming Service and Track Service are outlined below. Initially, the only application running is the ORBIX daemon, which handles requests for the clients and servers.

- The DICE Track Server submits a query for the naming service
- The name server is started
- The DICE Track Server finds the naming service through the server's Object Request Broker (ORB)
- The Server registers its service with the naming service

The naming service now has a reference to the DICE Track Service. The service itself can shut down, allowing system resources to be freed. The server will also shut down following inactivity for a pre-determined timeout period, however the ORBIX daemon remains active to respond to client requests. The sequence of steps that take place for client-server communications are as follows:

- The OpenMap Client requests a named service (the DICE Track Service)
- The DICE Track Service is started by the ORBIX daemon
- The naming service returns a reference to this service

The client can now communicate with the track service via the ORB. The OpenMap client, DICE Track Service and Naming Service all have their own ORB through which all communications occur.

Figure A2 shows a number of features of OpenMAP communicating with the track service. The pull down 'Layers' menu (see top left in Figure A2) shows the selection of services available via the naming service, services which include geospatial and track information. When a service is selected the data available at that service is displayed as a layer in the OpenMAP window. First a world map may be selected and displayed in order to provide context to the track information – the Northern Australia region is shown. Then various track services are selected (one at a time) including VOP, GBR and JORN to display a composite picture. The `TrackCollectionFactory` interface of the IDL is used to display the tracks. In Figure A2 the individual track points can be seen as well as the track (a line which connects points pertaining to the same target). White points/lines represent neutral forces (eg commercial aircraft), blue points/lines represent friendly forces, and red points/lines represent enemy forces. Finally each track object (be it a track or a track point) can be interrogated by clicking it. This accesses either the `Track` or `TrackPoint` interface of the IDL and exposes more information on the track object (for example the track ID, posture, and heading). This causes a new window to be displayed in OpenMap, see the bottom left corner of Figure A2.

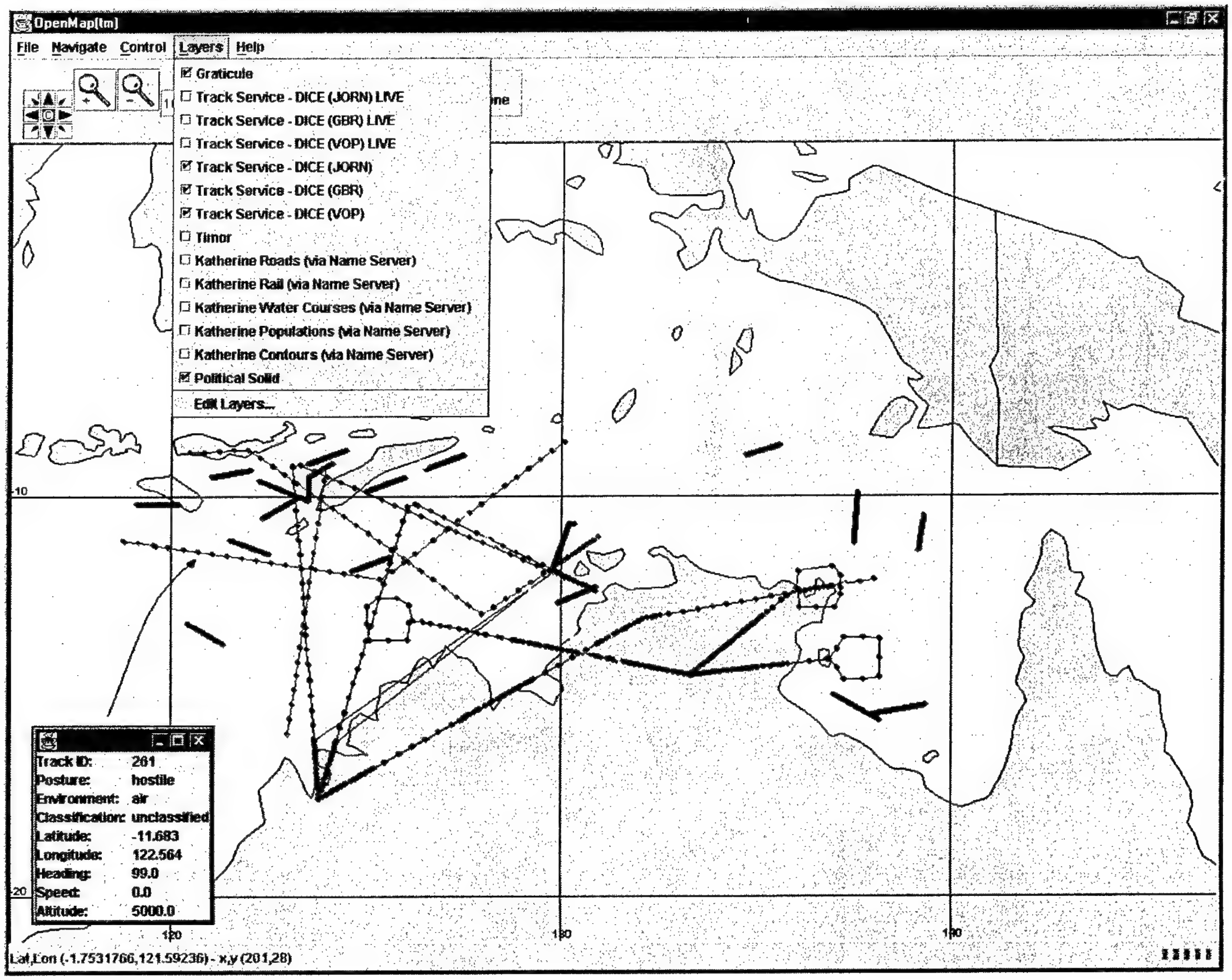


**Figure A2: The OpenMap client displaying data via the EXC3ITE track service.**

## A3 HLA EXC3ITE track federate and object model

An HLA EXC3ITE Track Federate has been developed that will enable the EXC3ITE Track Services to participate in a HLA federation. Instances of the federate provide tracks from numerous synthetic and real sources, using live or pre-recorded feeds, encapsulated within an appropriate SOM. A critical part of this development is to maximise reuse of the existing Track Service capabilities described earlier, specifically to leverage from the design effort and products of the Track Service CORBA IDL.

Figure A3 shows the resultant SOM that is a 1-1 mapping from the CORBA IDL (as shown in Figure A1). The HLA EXC3ITE Track Federate is also configured to support the 'Entity-State' subset of the Real-time Platform-level Reference FOM (RPR-FOM). The DICE federate is also able to communicate via these object models.

Development of the HLA EXC3ITE Track Service federate and its use within a resultant federation that is a hybrid of EXC3ITE and simulation products, provides a key vehicle for exploring and exploiting synergy between EXC3ITE and the use of HLA. That is, synergy between simulation and real systems.

### Object Class Structure Table

The Object Class Structure Table uses the following properties to categorise each class:

P = Publish – federate is capable of publishing the class.
S = Subscribe – federate is capable of subscribing to the class.
N = Neither – federate is not capable of publishing the class or subscribing to the class.

| Class 1 | Class2 | Class3 | Class4 |
|---|---|---|---|
| Track (P) | | | |
| BaseEntity (N) | PhysicalEntity (N) | Platform (N) | Aircraft (P) |
| | | | AmphibiousVehicle (P) |
| | | | GroundVehicle (P) |
| | | | MultiDomainPlatform (P) |
| | | | Spacecraft (P) |
| | | | SubmersibleVessel (P) |
| | | | SurfaceVessel (P) |

### Object Interaction Table

The Object Interaction Table uses the following properties to categorise each interaction:

- I = Initiates – federate is capable of initiating and sending the interaction.
- R = Reacts – federate is capable of subscribing and properly reacting to the interaction.
- S = Senses – federate is capable of subscribing to the interaction and utilising the interaction information.

| Interaction 1 | Interaction 2 |
|---|---|
| SimStateChange (R) | |

**Attribute Table**

The following properties apply for all attributes of all objects.

1.FED Delivery Category = "best_effort".
2.FED Message Ordering = "receive".

| Object | Attribute | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition | Update Type | Update Condition | Transferable / Acceptable | Updateable / Reflectable | Routing Space |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BaseEntity | AccelerationVector | AccelerationVectorStruct | 1 | N/A | N/A | N/A | N/A | Conditional | AccelerationChange | N | U | N/A |
| | AngularVelocityVector | AngularVelocityVectorStruct | 1 | N/A | N/A | N/A | N/A | Conditional | AngVelocityChange | N | U | N/A |
| | DeadReckoningAlgorithm | DeadReckoningAlgorithmEnum8 | 1 | N/A | N/A | N/A | N/A | Conditional | On change | N | U | N/A |
| | EntityType | EntityTypeStruct | 1 | N/A | N/A | N/A | N/A | Static | N/A | N | U | N/A |
| | EntityIdentifier | EntityIdentifierStruct | 1 | N/A | N/A | N/A | N/A | Static | N/A | N | U | N/A |
| | IsFrozen | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | Orientation | OrientationStruct | 1 | N/A | N/A | N/A | N/A | Conditional | OrientationChange | N | U | N/A |
| | WorldLocation | WorldLocationStruct | 1 | N/A | N/A | N/A | N/A | Conditional | LocationChange | N | U | N/A |
| | VelocityVector | VelocityVectorStruct | 1 | N/A | N/A | N/A | N/A | Conditional | VelocityChange | N | U | N/A |
| PhysicalEntity | AlternateEntityType | EntityTypeStruct | 1 | N/A | N/A | N/A | N/A | Static | N/A | N | U | N/A |
| | ArticulatedParametersArray | ArticulatedParameterStruct | 0+ | N/A | N/A | N/A | N/A | Conditional | On change | N | U | N/A |

| Object | Attribute | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition | Update Type | Update Condition | Transferable / Acceptable | Updateable / Reflectable | Routing Space |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CamouflageType | CamouflageEnum32 | 1 | N/A | N/A | N/A | N/A | Conditional | On change | N | U | N/A |
| | DamageState | DamageStatusEnum32 | 1 | N/A | N/A | N/A | N/A | Conditional | On change | N | U | N/A |
| | EngineSmokeOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | FirePowerDisabled | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | FlamesPresent | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | ForceIdentifier | ForceIdentifierEnum8 | 1 | N/A | N/A | N/A | N/A | Conditional | On change | N | U | N/A |
| | HasAmmunitionSupplyCap | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Static | N/A | N | U | N/A |
| | HasFuelSupplyCap | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Static | N/A | N | U | N/A |
| | HasRecoveryCap | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Static | N/A | N | U | N/A |
| | HasRepairCap | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Static | N/A | N | U | N/A |
| | Immobilized | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | IsConcealed | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | Marking | MarkingStruct | 1 | N/A | N/A | N/A | N/A | Static | N/A | N | U | N/A |

| Object | Attribute | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition | Update Type | Update Condition | Transferable / Acceptable | Updateable / Reflectable | Routing Space |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PowerPlantOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | SmokePlumePresent | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | TentDeployed | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | TrailingEffectsCode | TrailingEffectsCodeEnum32 | 1 | N/A | N/A | N/A | N/A | Conditional | On change | N | U | N/A |
| Platform | AfterburnerOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | AntiCollisionLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | BlackOutBrakeLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | BlackOutLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | BrakeLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | FormationLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | HatchState | HatchStateEnum32 | 1 | N/A | N/A | N/A | N/A | Conditional | On change | N | U | N/A |
| | HeadLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |

| Object | Attribute | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition | Update Type | Update Condition | Transferable / Acceptable | Updateable / Reflectable | Routing Space |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | InteriorLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | LandingLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | LauncherRaised | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | NavigationLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | RampDeployed | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | RunningLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | SpotLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| | TailLightsOn | boolean | 1 | TRUE/ FALSE | N/A | perfect | perfect | Conditional | On change | N | U | N/A |
| Track | TrackId | string | 1 | | | perfect | always | Conditional | | N | U | N/A |
| | security | SecurityLabelStruct | 1 | N/A | N/A | N/A | N/A | Conditional | | N | U | N/A |
| | trackpoints | TrackPointStruct | 1+ | N/A | N/A | N/A | N/A | Conditional | | N | U | N/A |
| | NumTrackpoints | unsigned long | 1 | | | perfect | always | Conditional | | N | U | N/A |
| | updateSegment | TrackPointStruct | 0+ | N/A | N/A | N/A | N/A | Conditional | | N | U | N/A |
| | NumUpdateTrackpoints | unsigned long | 1 | | | perfect | always | Conditional | | N | U | N/A |

| Object | Attribute | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition | Update Type | Update Condition | Transferable / Acceptable | Updateable / Reflectable | Routing Space |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Annotation | string | 1 | | | perfect | always | Conditional | | N | U | N/A |
| | Simulated | boolean | 1 | | | perfect | always | Conditional | | N | U | N/A |

## Parameter Table

| Interaction | Parameter | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition | Routing Space |
|---|---|---|---|---|---|---|---|---|
| SimStateChange | NewState | SimStateEnum | 1 | N/A | N/A | N/A | N/A | N/A |
| | SimTime | DICETime | 1 | N/A | N/A | N/A | N/A | |
| | SimRate | float | 1 | | | perfect | always | |
| | WallClockTime | DICETime | 1 | N/A | N/A | N/A | N/A | |

## Complex Datatype Table

| Complex Datatype | Field Name | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition |
|---|---|---|---|---|---|---|---|
| AccelerationVectorStruct | XAcceleration | float | 1 | m/s/s | | perfect | always |
| | YAcceleration | float | 1 | m/s/s | | perfect | always |
| | ZAcceleration | float | 1 | m/s/s | | perfect | always |
| AngularVelocityVectorStruct | XAngularVelocity | float | 1 | radians/s | | perfect | always |
| | YAngularVelocity | float | 1 | radians/s | | perfect | always |
| | ZAngularVelocity | float | 1 | radians/s | | perfect | always |
| ArticulatedParameterStruct | ArticulatedParameterChange | octet | 1 | N/A | 1 | perfect | always |
| | Padding | octet | 1 | N/A | N/A | perfect | always |
| | PartAttachedTo | unsigned short | 1 | N/A | 1 | perfect | always |
| | ParameterValue | any | 1 | | | perfect | always |
| ArticulatedPartsStruct | Class | ArticulatedPartsTypeEnum32 | 1 | N/A | N/A | N/A | N/A |
| | TypeMetric | ArticulatedTypeMetricEnum32 | 1 | N/A | N/A | N/A | N/A |
| | Value | float | 1 | | | perfect | always |
| AttachedPartsStruct | Station | StationEnum32 | 1 | N/A | N/A | N/A | N/A |
| | StoreType | EntityTypeStruct | 1 | N/A | N/A | N/A | N/A |
| DICETime | Seconds | unsigned long | 1 | s | | perfect | always |
| | MilliSeconds | unsigned short | 1 | ms | | perfect | always |
| DistanceStruct | Value | float | 1 | | | perfect | always |

| Complex Datatype | Field Name | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition |
|---|---|---|---|---|---|---|---|
| | Units | DistanceUnitsEnum | 1 | N/A | N/A | N/A | N/A |
| EntityIdentifierStruct | FederateIdentifier | FederateIdentifierStruct | 1 | N/A | N/A | N/A | N/A |
| | EntityNumber | unsigned short | 1 | N/A | 1 | perfect | always |
| EntityTypeStruct | EntityKind | octet | 1 | N/A | 1 | perfect | always |
| | Domain | octet | 1 | N/A | 1 | perfect | always |
| | CountryCode | unsigned short | 1 | N/A | 1 | perfect | always |
| | Category | octet | 1 | N/A | 1 | perfect | always |
| | Subcategory | octet | 1 | N/A | 1 | perfect | always |
| | Specific | octet | 1 | N/A | 1 | perfect | always |
| | Extra | octet | 1 | N/A | 1 | perfect | always |
| ErrorEllipseStruct | LenSemiMajor | double | 1 | nm | | perfect | always |
| | LenSemiMinor | double | 1 | nm | | perfect | always |
| | BearingMajor | double | 1 | degt | | perfect | always |
| FederateIdentifierStruct | SiteID | unsigned short | 1 | N/A | N/A | perfect | always |
| | ApplicationID | unsigned short | 1 | N/A | N/A | perfect | always |
| IffStruct | Mode1 | string | 4 | | | perfect | always |
| | Mode2 | string | 4 | | | perfect | always |
| | Mode3 | string | 4 | | | perfect | always |

| Complex Datatype | Field Name | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition |
|---|---|---|---|---|---|---|---|
| | Mode4 | string | 4 | | | perfect | always |
| MarkingStruct | MarkingEncodingType | MarkingEncodingEnum8 | 1 | N/A | N/A | N/A | N/A |
| | MarkingData | octet | 11 | – | | perfect | always |
| OrientationStruct | Psi | float | 1 | radians | | perfect | always |
| | Theta | float | 1 | radians | | perfect | always |
| | Phi | float | 1 | radians | | perfect | always |
| ParameterValueStruct | ArticulatedParameterType | ParameterTypeEnum32 | 1 | N/A | N/A | N/A | N/A |
| | ArticulatedParts | ArticulatedPartsStruct | 0-1 | N/A | N/A | N/A | N/A |
| | AttachedParts | AttachedPartsStruct | 0-1 | N/A | N/A | N/A | N/A |
| PointStruct | X | double | 1 | | | perfect | always |
| | Y | double | 1 | | | perfect | always |
| SecurityLabelStruct | SecClass | ClassificationEnum | 1 | N/A | N/A | N/A | N/A |
| | CaveatList | string | 0+ | | | perfect | always |
| | ReleasabilityList | string | 0+ | | | perfect | always |
| SpeedStruct | Value | float | 1 | | | perfect | always |
| | Units | SpeedUnitsEnum | 1 | N/A | N/A | N/A | N/A |
| TimeStruct | Year | unsigned short | 1 | | | perfect | always |
| | Month | unsigned short | 1 | | | perfect | always |

| Complex Datatype | Field Name | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition |
|---|---|---|---|---|---|---|---|
| | Day | unsigned short | 1 | | | perfect | always |
| | Hour | unsigned short | 1 | | | perfect | always |
| | Minute | unsigned short | 1 | | | perfect | always |
| | Second | unsigned short | 1 | | | perfect | always |
| | Millisecond | unsigned short | 1 | | | perfect | always |
| TrackPointStruct | Position | PointStruct | 1 | N/A | N/A | N/A | N/A |
| | Security | SecurityLabelStruct | 1 | N/A | N/A | N/A | N/A |
| | Posture | PostureEnum | 1 | N/A | N/A | N/A | N/A |
| | ErrorEllipse | ErrorEllipseStruct | 0+ | N/A | N/A | N/A | N/A |
| | Altitude | DistanceStruct | 0+ | N/A | N/A | N/A | N/A |
| | Depth | DistanceStruct | 0+ | N/A | N/A | N/A | N/A |
| | Speed | SpeedStruct | 0+ | N/A | N/A | N/A | N/A |
| | CourseDegt | double | 0+ | | | perfect | always |
| | Callsign | string | 0+ | | | perfect | always |
| | Iff | IffStruct | 0+ | N/A | N/A | N/A | N/A |
| | Env | OpEnvEnum | 1 | N/A | N/A | N/A | N/A |
| | Time | TimeStruct | 1 | N/A | N/A | N/A | N/A |
| | Padding | string | 2 | N/A | N/A | perfect | always |

| Complex Datatype | Field Name | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition |
|---|---|---|---|---|---|---|---|
| VelocityVectorStruct | XVelocity | float | 1 | m/s | | perfect | always |
| | YVelocity | float | 1 | m/s | | perfect | always |
| | ZVelocity | float | 1 | m/s | | perfect | always |
| WorldLocationStruct | X | double | 1 | meters | | perfect | always |
| | Y | double | 1 | meters | | perfect | always |
| | Z | double | 1 | meters | | perfect | always |

## Enumerated Datatype Table

(Abbreviated for clarity.)

| Identifier | Enumerator | Representation |
|---|---|---|
| ArticulatedPartsTypeEnum32 | Other | 0 |
| | ... | ... |
| ArticulatedTypeMetricEnum32 | Position | 1 |
| | ... | ... |
| CamouflageEnum32 | UniformPaintScheme | 0 |
| | ... | ... |
| ClassificationEnum | Unclassified | 1 |
| | Restricted | 2 |
| | Confidential | 3 |
| | Secret | 4 |
| | Topsecret | 5 |
| DamageStatusEnum32 | NoDamage | 0 |
| | ... | ... |
| DeadReckoningAlgorithmEnum8 | Other | 0 |
| | ... | ... |
| DistanceUnitsEnum | M | 1 |
| | Ft | 2 |
| ForceIdentifierEnum8 | Other | 0 |
| | ... | ... |
| HatchStateEnum32 | NotApplicable | 0 |
| | ... | ... |
| MarkingEncodingEnum8 | Other | 0 |
| | ... | ... |
| OpEnvEnum | SubSurface | 1 |
| | Surface | 2 |
| | Land | 3 |
| | Air | 4 |
| | Space | 5 |
| ParameterTypeEnum32 | ArticulatedPart | 0 |
| | ... | ... |
| PostureEnum | Unknown | 1 |
| | Friend | 2 |

| Identifier | Enumerator | Representation |
|---|---|---|
| | Neutral | 3 |
| | Hostile | 4 |
| | Suspect | 5 |
| | Faker | 6 |
| | Bogey | 7 |
| SimStateEnum | Start | 1 |
| | Stop | 2 |
| | Pause | 3 |
| | Resume | 4 |
| SpeedUnitsEnum | Kts | 1 |
| | Kph | 2 |
| | Mph | 3 |
| StationEnum32 | Nothing_Empty | 0 |
| | ... | ... |
| TrailingEffectsCodeEnum32 | NoTrail | 0 |
| | ... | ... |

**Figure A3: HLA EXC3ITE Track Federate SOM**

# Appendix B: The High Level Architecture (HLA)

As mentioned in Section 3 HLA is composed of three interrelated elements, namely the HLA Rules, the HLA Interface Specification, and the HLA Object Model Template (OMT). The HLA Rules define how federates and federations are built, the Interface Specification defines how federates (and therefore federations) interact with the Run-time Infrastructure, and the OMT is a way of documenting structural information about federates and federations. This section will describe these three elements in more detail and the three types of compliance associated with HLA.

## B1 HLA Rules

The HLA Rules document the general principles underpinning the development of HLA. There are a total of ten rules, with five referring to simulation federations, and five pertaining to component federates. They are listed below:

1. Federations shall have an HLA FOM, documented in accordance with the OMT.
2. In a federation, all representation of objects in the FOM shall be within the federates, not in the RTI.
3. During federation execution, the exchange of FOM data between federates shall occur exclusively via the RTI.
4. During federation execution, all federates shall interact with the RTI in accordance with the HLA Interface Specification.
5. During a federation execution, only one federate shall own any given attribute of any particular object at any moment.
6. Federates shall have an HLA SOM, documented in accordance with the OMT.
7. Federates shall be able to update and/or reflect any attributes of objects in their SOM and send and/or receive SOM object interactions externally, as specified in their SOM.
8. Federates shall be able to transfer and/or accept ownership of attributes dynamically during a federation execution, as specified in their SOM.
9. Federates shall be able to vary the conditions under which they provide updates of attributes of objects, as specified in their SOM.
10. Federates shall be able to manage local time in a way which will allow them to coordinate data exchange with other members of the federation.

## B2 Components of the HLA Object Model Template

The OMT specifies that HLA object models be documented in the form of a number of tables that specify information about classes of objects, their attributes and their interactions. The OMT is the standard framework for specifying both Simulation Object Models (SOM) and Federation Object Models (FOM). The OMT consists of seven different tables, and these are listed briefly below.

1. Object Model Identification Table

   The Object Model Identification Table provides key identifying information about the federation or federates.

2. Object Class Table

   This table contains class-subclass hierarchy of the object classes.

3. Interaction Class Table

   Actions taken by an object in one federate that may have an effect on objects in a different federate are defined in the Interaction Class Table.

4. Attribute Table

   Each class is characterised by a fixed set of attribute types. These attributes are named portions of the state of their object whose values can change over time.

5. Parameter Table

   The Parameter Table contains the full set of parameters associated with the interactions specified in the Interaction Class Table.

6. Routing Space

   In order to enable and limit the flow of object attributes and interactions between federates the Routing Space Table provides a common framework for specifying the data distribution model.

7. FOM/SOM Lexicon

   The FOM/SOM Lexicon provides semantic information to aid in understanding the information contained in the OMT.

It is worth noting that the OMT does not correspond entirely with the standard definitions commonly seen in other object-oriented development methodologies. For instance, the HLA object models do not include the object operations of OO static models. The dynamic component of the HLA object model does not include the event sequences and transition models, preferring instead

to focus exclusively on pairwise interactions. Furthermore, multiple inheritance is not supported in an HLA object class hierarchy.

The Object Model Development Tool (OMDT), available from the DMSO, automates the process of constructing SOMs and FOMs that are in accordance with the OMT. The OMDT can then generate the FED file that is required by the RTI for federation execution.

## B3 Interface Specification

The Interface Specification is a generic specification for the various language-specific Application Programming Interfaces (API); it defines the functional interfaces between any component federate and the services provided by the RTI. These services fall into six categories: Federation Management, Declaration Management, Object Management, Ownership Management, Time Management, and Data Distribution Management.

Federation Management refers to functions relating to the creation, deletion, modification, and dynamic control of the execution of an entire federation. Declaration Management services provide federates with a mechanism to declare to the RTI their interest in particular object state information, as well as interactions that the federate generates and receives. Object Management refers to services dealing with the creation, deletion, and modification of the objects themselves. Ownership Management services permit one federate to transfer ownership of object attributes to another. The federate that owns a particular attribute of an object may assign new values to that attribute. Furthermore, a predefined attribute exists for each object that gives the owning federate the right to delete that object. Time Management services facilitate the coordination of logical time advancement throughout the federation. Data Distribution Management is the set of services to facilitate the explicit management of how data is distributed to federates. While declaration management services are used to specify which types of data a given federate will send or receive, data distribution services are used to define the specific conditions under which data values will be provided or expected.

## B4 HLA Compliance

There are three types of compliance associated with HLA. These are federate, federation and RTI compliance.

For a federate to be HLA compliant it must satisfy a compliance checklist for federates. The federate compliance checklist includes the following:

- the federate shall have a SOM in accordance with the OMT
- be able to update and/or reflect attributes in its SOM
- send and/or receive interactions in its SOM
- transfer and/or accept ownership of attributes in its SOM

- shall specify its time management mechanism (eg time regulating and/or time constrained)
- interact with the RTI in accordance with the HLA interface specification
- A federate can also be submitted to DMSO for federate compliance testing.

A similar checklist exists for a federation to be HLA compliant. This checklist includes the following:

- the federation shall have a FOM in accordance with the OMT
- all objects are represented in federates
- all FOM data is interchanged through the RTI
- federates shall interact with the RTI in accordance with the HLA interface specification
- attributes shall be owned by only one federate at any time

There also exists a compliance checklist for RTI. This checklist is only necessary if you are developing your own RTI.

# Appendix C: Federation Development

When developing a High Level Architecture (HLA) federation all the participating federates must have a defined Simulation Object Model (SOM). In our case it was necessary to decide the make-up of the SOMs for each federate (DICE, HLA Track Federate and HLA Messaging Federate). The Federation Object Model (FOM) is a combination of the SOMs from each participating federate. During development of the FOM it is also necessary to determine which federates will publish and/or subscribe to objects and interactions within the FOM. This appendix will describe the development of the SOM for each federate, federate compliance level, federate development and the final Federation Object Model.

## C1 An HLA-enabled version of the DICE simulation software suite

As described in section 5.2.1 the main choice during the development of the SOM for the DICE simulation was whether to describe the formatted messages as either objects or interactions. Interactions were selected. Other decisions during SOM development included whether to describe the formatted messages individually, of which there are many, or provide a generic representation. A generic representation was chosen as describing the individual formatted messages within the SOM would have been cumbersome and a duplication of the DMDD. Also it was decided to describe the formatted messages in their completed formatted form along with the structure of the message.

The resultant DICE SOM is described in the following table. Receiving federates can either utilise the DICEForm (ADFORM like formatted messages) as a whole or can delve deeper to retrieve certain information. In return other federates can send a DICEForm as a whole for the HLA PUI to forward to other C3I nodes within the DICE simulation or they can provide certain information contained within the DICEForm and the HLA PUI can fill in the rest. Using this DICE SOM testing can easily be done by having two HLA PUIs executing and sending interactions to each other. Also this SOM provides the ability to connect to CSS that may already have the ability to construct military formatted messages, such as ADFORM, within their own system.

**Object Model Identification Table**

| Category | Information |
|---|---|
| Name | DICE SOM |
| Version | 1.0 |
| Date | 01/11/2000 |
| Purpose | DICE SOM for use with the DICE Simulation Software Suite. |
| Application Domain | C3I |
| Sponsor | EXC3ITE |
| POC (Title, First, Last) | Dr SPOC |
| POC Organization | SSA/ITD/DSTO |

| Category | Information |
|---|---|
| POC Telephone | (+61) 8 ### #### |
| POC Email | Dr.SPOC@dsto.defence.gov.au |

### Object Class Structure Table

The DICE SOM does not use objects.

### Object Interaction Table

The Object Interaction Table uses the following properties to categorise each interaction:

I = Initiates – federate is capable of initiating and sending the interaction.
R = Reacts – federate is capable of subscribing and properly reacting to the interaction.
S = Senses – federate is capable of subscribing to the interaction and utilising the interaction information.

| Interaction 1 | Interaction 2 |
|---|---|
| DICEMessage (IR) | - |
| SimStateChange (IR) | - |

### Attribute Table

The DICE SOM does not use object attributes.

### Parameter Table

| Interaction | Parameter | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition | Routing Space |
|---|---|---|---|---|---|---|---|---|
| DICEMessage | Source | string | 1 | | | perfect | always | N/A |
| | Destination | string | 1 | | | perfect | always | |
| | Link | string | 1 | | | perfect | always | |
| | TimeSent | DICETime | 1 | N/A | N/A | N/A | N/A | |
| | TimeReceived | DICETime | 1 | N/A | N/A | N/A | N/A | |
| | DICEForm | DICEForm | 1 | N/A | N/A | N/A | N/A | |
| SimStateChange | NewState | SimStateEnum | 1 | N/A | N/A | N/A | N/A | N/A |
| | SimTime | DICETime | 1 | N/A | N/A | N/A | N/A | |
| | SimRate | float | 1 | | | perfect | always | |
| | WallClockTime | DICETime | 1 | N/A | N/A | N/A | N/A | |

### Complex Datatype Table

| Complex Datatype | Field Name | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition |
|---|---|---|---|---|---|---|---|

| Complex Datatype | Field Name | Datatype | Cardinality | Units | Resolution | Accuracy | Accuracy Condition |
|---|---|---|---|---|---|---|---|
| DICEForm | Abbreviation | string | 1 | | | perfect | always |
| | Occurrence | string | 1 | | | perfect | always |
| | DICEFormString | string | 1 | | | perfect | always |
| | SetList | DICESet | 0+ | N/A | N/A | N/A | N/A |
| DICESet | Abbreviation | string | 1 | | | perfect | always |
| | FieldList | DICEField | 0+ | N/A | N/A | N/A | N/A |
| DICEField | Mapping | string | 1 | | | perfect | always |
| | Data | string | 1 | | | perfect | always |
| DICETime | Seconds | unsigned long | 1 | s | | perfect | always |
| | MilliSeconds | unsigned short | 1 | ms | | perfect | always |

### Enumerated Datatype Table

| Identifier | Enumerator | Representation |
|---|---|---|
| SimStateEnum | Start | 1 |
| | Stop | 2 |
| | Pause | 3 |
| | Resume | 4 |

### Routing Space Table

The DICE SOM does not use routing space.

### Class Lexicon

The DICE SOM does not use objects.

### Interaction Lexicon

| Term | Definition |
|---|---|
| DICEMessage | Represents a DICEMessage as used in the DICE Simulation Software Suite. |
| SimStateChange | Simulation state change. |

### Attribute Lexicon

The DICE SOM does not use object attributes.

## Parameter Lexicon

| Interaction | Term | Definition |
|---|---|---|
| DICEMessage | Source | Source of the DICE Message. |
| | Destination | Destination of the DICE Message. |
| | Link | Name of the link that the DICE Message was sent along. |
| | TimeSent | Time at which the DICE Message was sent. |
| | TimeReceived | Time at which the DICE Message was received. |
| | DICEForm | The DICE Form or ADFORM. |
| SimStateChange | NewState | New simulation state. |
| | SimTime | Current simulation time. |
| | SimRate | Current simulation rate. |
| | WallClockTime | Current wall clock or system time. |

## Complex Datatype Lexicon

| Complex Datatype | Definition |
|---|---|
| DICEForm | A DICE Form (or ADFORM). |
| DICESet | A DICE Form set. |
| DICEField | A DICE Form field. |
| DICETime | DICE time. |

## Complex Datatype Field Lexicon

| Complex Datatype | Field Name | Definition |
|---|---|---|
| DICEForm | Abbreviation | DICE Form abbreviation. |
| | Occurrence | DICE Form occurrence name. |
| | DICEFormString | DICE Form as a slash deliminated string. |
| | SetList | A list of sets. |
| DICESet | Abbreviation | Set abbreviation. |
| | FieldList | A list of fields. |
| DICEField | Mapping | Field mapping. |
| | Data | Field data. |
| DICETime | Seconds | Number of seconds in absolute time or delta. If the Time holds an absolute time (rather than a delta) this will be the number of seconds since 1 Jan 1970 (UTC). |
| | MilliSeconds | Number of milliseconds (less than 1000 and non-negative). |

Apart from developing the DICE HLA PUI to handle the DICE SOM it was necessary to determine which services, apart from the mandatory ones, of the interface specification it would handle. This is detailed in the tables below. A minimal handling means either the DICE HLA PUI accepts a RTI initiated service but no action is performed or responds to a RTI initiated service through the calling of a service. An example of this is the handling of the 'Initiate Federate Save' service that the DICE HLA PUI accepts but no save is done. The DICE HLA PUI responds to this by calling 'Federate Save Begun', and so on. This is done so that the federation can continue with the save process.

Service Requirements

* - Indicates mandatory service
† - RTI-initiated service; federate must accept service invocation

**Federation Management**

| Service | IF Ref | OMT Ref | Prog Ref | DICE |
|---|---|---|---|---|
| Create Federation Execution * | 4.2 | | A.1.1 | Yes |
| Destroy Federation Execution * | 4.3 | | A.1.2 | Yes |
| Join Federation Execution * | 4.4 | | A.1.10 | Yes |
| Resign Federation Execution * | 4.5 | | A.1.18 | Yes |
| Register Federation Synchronization Point | 4.6 | | A.1.12 | Yes |
| Confirm Synchronisation Point Registration †<br>Synchronization Point Registration Succeeded †<br>Synchronization Point Registration Failed † | 4.7 | | <br>B.1.16<br>B.1.15 | Yes |
| Announce Synchronisation Point † | 4.8 | | B.1.1 | Yes |
| Synchronisation Point Achieved | 4.9 | | A.1.22 | Yes |
| Federation Synchronised † | 4.10 | | B.1.7 | Yes |
| Request Federation Save | 4.11 | | A.1.14 | No |
| Initiate Federate Save † | 4.12 | | B.1.9 | Minimal |
| Federate Save Begun | 4.13 | | A.1.5 | Minimal |
| Federate Save Complete<br>Federate Save Not Complete | 4.14 | | A.1.7<br>A.1.9 | Minimal |
| Federation Saved †<br>Federation Not Saved † | 4.15 | | B.1.6<br>B.1.3 | Minimal |
| Request Restore | 4.16 | | A.1.13 | No |
| Confirm Federation Restoration Request †<br>Request Federation Restore Failed †<br>Request Federation Restore Succeeded † | 4.17 | | <br>B.1.14<br>B.1.13 | Minimal |
| Federate Restore Begun † | 4.18 | | B.1.4 | Minimal |

| Service | IF Ref | OMT Ref | Prog Ref | DICE |
|---|---|---|---|---|
| Initiate Federate Restore † | 4.19 | | B.1.8 | Minimal |
| Federate Restore Complete<br>Federate Restore Not Complete | 4.20 | | A.1.3<br>A.1.4 | Minimal |
| Federation Restored †<br>Federation Not Restored † | 4.21 | | B.1.5<br>B.1.2 | Minimal |

## Declaration Management

| Service | IF Ref | OMT Ref | Prog Ref | DICE |
|---|---|---|---|---|
| Publish Object Class | 5.2 | 4.2.2 | A.2.2 | Yes |
| Unpublish Object Class | 5.3 | | A.2.7 | Yes |
| Publish Interaction Class | 5.4 | 4.3.2 | A.2.1 | Yes |
| Unpublish Interaction Class | 5.5 | | A.2.6 | Yes |
| Subscribe Object Class Attributes | 5.6 | 4.2.2 | A.2.5 | Yes |
| Unsubscribe Object Class | 5.7 | | A.2.9 | Yes |
| Subscribe Interaction Class | 5.8 | 4.3.2 | A.2.3 | Yes |
| Unsubscribe Interaction Class | 5.9 | | A.2.8 | Yes |
| Start Registration For Object Class † | 5.10 | | B.2.2 | Yes |
| Stop Registration For Object Class † | 5.11 | | B.2.5 | Yes |
| Turn Interactions On † | 5.12 | | B.2.8 | Yes |
| Turn Interactions Off † | 5.13 | | B.2.7 | Yes |

## Object Management

| Service | IF Ref | OMT Ref | Prog Ref | DICE |
|---|---|---|---|---|
| Register Object Instance | 6.2 | 4.2.2 | A.3.7 | Yes |
| Discover Object Instance † | 6.3 | 4.2.2 | B.3.4 | Yes |
| Update Attribute Values | 6.4 | 4.4.2 | A.3.12 | Yes |
| Reflect Attribute Values † | 6.5 | 4.4.2 | B.3.7 | Yes |
| Send Interaction | 6.6 | 4.3.2 | A.3.11 | Yes |
| Receive Interaction † | 6.7 | 4.3.2 | B.3.6 | Yes |
| Delete Object Instance | 6.8 | | A.3.4 | Yes |
| Remove Object Instance † | 6.9 | | B.3.10 | Yes |
| Local Delete Object Instance | 6.10 | | A.3.5 | No |
| Change Attribute Transportation Type | 6.11 | | A.3.1 | No |
| Change Interaction Transportation Type | 6.12 | | A.3.2 | No |
| Attributes In Scope † | 6.13 | | B.3.1 | Minimal |

| Service | IF Ref | OMT Ref | Prog Ref | DICE |
|---|---|---|---|---|
| Attributes Out Of Scope † | 6.14 | | B.3.2 | Minimal |
| Request Object Attribute Value Update<br>Request Class Attribute Value Update<br>Request Object Value Update | 6.15 | | <br>A.3.8<br>A.3.10 | Yes |
| Provide Attribute Value Update † | 6.16 | 4.3.2 | B.3.5 | Yes |
| Turn Updates On For Object Instance † | 6.17 | | B.3.11 | Yes |
| Turn Updates Off For Object Instance † | 6.18 | | B.3.12 | Yes |

## Ownership Management

| Service | IF Ref | OMT Ref | Prog Ref | DICE |
|---|---|---|---|---|
| Unconditional Attribute Ownership Divestiture | 7.2 | 4.4.2 | A.4.12 | No |
| Negotiated Attribute Ownership Divestiture | 7.3 | 4.4.2 | A.4.8 | No |
| Request Attribute Ownership Assumption † | 7.4 | 4.4.2 | B.4.8 | Minimal |
| Attribute Ownership Divestiture Notification † | 7.5 | 4.4.2 | B.4.4 | Minimal |
| Attribute Ownership Acquisition Notification † | 7.6 | 4.4.2 | B.4.3 | Minimal |
| Attribute Ownership Acquisition | 7.7 | 4.4.2 | A.4.2 | No |
| Attribute Ownership Acquisition If Available | 7.8 | 4.4.2 | A.4.3 | No |
| Attribute Ownership Unavailable † | 7.9 | 4.4.2 | B.4.5 | Minimal |
| Request Attribute Ownership Release † | 7.10 | 4.4.2 | B.4.9 | Minimal |
| Attribute Ownership Release Response | 7.11 | 4.4.2 | A.4.4 | Minimal |
| Cancel Negotiated Attribute Ownership Divestiture | 7.12 | | A.4.6 | No |
| Cancel Attribute Ownership Acquisition | 7.13 | | A.4.5 | No |
| Confirm Attribute Ownership Acquisition Cancellation † | 7.14 | | B.4.6 | Minimal |
| Query Attribute Ownership | 7.15 | | A.4.9 | No |
| Inform Attribute Ownership †<br>Attribute Is Not Owned †<br>Attribute Is Owned By RTI † | 7.16 | | B.4.7<br>B.4.1<br>B.4.2 | Minimal |
| Is Attribute Owned by Federate | 7.17 | | A.4.7 | No |

## Time Management

| Service | IF Ref | OMT Ref | Prog Ref | DICE |
|---|---|---|---|---|
| Enable Time Regulation | 8.2 | | A.5.8 | No |
| Time Regulation Enabled † | 8.3 | | B.5.4 | Minimal |
| Disable Time Regulation | 8.4 | | A.5.5 | No |
| Enable Time Constrained | 8.5 | | A.5.7 | No |

| Service | IF Ref | OMT Ref | Prog Ref | DICE |
|---|---|---|---|---|
| Time Constrained Enabled † | 8.6 | | B.5.3 | Minimal |
| Disable Time Constrained | 8.7 | | A.5.4 | No |
| Time Advance Request | 8.8 | | A.5.25 | Yes |
| Time Advance Request Available | 8.9 | | A.5.26 | No |
| Next Event Request | 8.10 | | A.5.11 | No |
| Next Event Request Available | 8.11 | | A.5.12 | No |
| Flush Queue Request | 8.12 | | A.5.9 | No |
| Time Advance Grant † | 8.13 | | B.5.2 | Minimal |
| Enable Asynchronous Delivery | 8.14 | | A.5.6 | No |
| Disable Asynchronous Delivery | 8.15 | | A.5.3 | No |
| Query LBTS | 8.16 | | A.5.14 | Yes |
| Query Federate Time | 8.17 | | A.5.13 | Yes |
| Query Min Next Event Time | 8.18 | | A.5.16 | No |
| Modify Lookahead | 8.19 | | A.5.10 | Yes |
| Query Lookahead | 8.20 | | A.5.15 | Yes |
| Retract | 8.21 | | A.5.22 | No |
| Request Retraction † | 8.22 | | B.5.1 | Minimal |
| Change Attribute Order Type | 8.23 | | A.5.1 | No |
| Change Interaction Order Type | 8.24 | | A.5.2 | No |

## Data Distribution Management

| Service | IF Ref | OMT Ref | Prog Ref | DICE |
|---|---|---|---|---|
| Create Region | 9.2 | | A.6.2 | No |
| Modify Region | 9.3 | | A.6.4 | No |
| Delete Region | 9.4 | | A.6.3 | No |
| Register Object Instance With Region | 9.5 | 4.4.2 | A.6.5 | No |
| Associate Region For Updates | 9.6 | 4.4.2 | A.6.1 | No |
| Unassociate Region For Updates | 9.7 | | A.6.10 | No |
| Subscribe Object Class Attributes With Region | 9.8 | 4.4.2 | A.6.9 | No |
| Unsubscribe Object Class With Region | 9.9 | | A.6.12 | No |
| Subscribe Interaction Class With Region | 9.10 | 4.5.2 | A.6.8 | No |
| Unsubscribe Interaction Class With Region | 9.11 | | A.6.11 | No |
| Send Interaction With Region | 9.12 | 4.5.2 | A.6.7 | No |
| Request Class Attribute Value Update With Region | 9.13 | | A.6.6 | No |

As specified in the Time Management services the DICE HLA PUI is neither Time Constrained nor Time Regulating. The best way to handle time between DICE and the RTI has yet to be determined. Also the DICE HLA PUI handles none of the services provided by Data Distribution Management.

During the design phase of the project it was decided that the DICE HLA PUI should have the ability to employ any FOM without the need for software engineering. This was achieved through adding those objects and/or interactions that the DICE HLA PUI wishes to publish and/or subscribe to to the DICE SOM. The SOM is then read by the DICE HLA PUI at start-up through software provided by the Virtual Ship Project. The software provides the ability to parse the OMT file, produced by the OMDT, and store the results. It also provides tools for automating the publish and subscribe features of HLA for federates and federations. Once the DICE SOM is read its objects and/or interactions can be converted to formatted messages through data-driven mappings. Vice-versa is also possible. Although handling any FOM was achievable problems arose when dealing with other federates running on different operating systems. Two problems arose, byte ordering of values and word alignment of complex types. The byte ordering issue was overcome by passing all values with big-endian byte ordering. This required the DICE HLA PUI, when executing on a PC operating system, to employ byte swapping.

The other issue required that alignment rules be taken into consideration when constructing complex types. Complex types shall be organised such that all base types (integers and floating point numbers) start on an offset which is a multiple of their own size. For example, the offset of a 32 bit float, within a complex type could be zero, 32, 64 or any other multiple of 32. Padding shall be added to the complex type if this internal alignment cannot be achieved through simple re-arrangement. All padding fields shall be set to zero. The following example illustrates this guidance: Using C syntax, we show two versions of a complex data type below:

```
struct BadType {
        char aChar ; /* 8 bits */
        short aShort ; /* 16 bits */
        long aLong ; /* 32 bits */
} ;
```

```
struct GoodType {
        long aLong ; /* 32 bits */
        short aShort ; /* 16 bits */
        char aChar ; /* 8 bits */
} ;
```

The "BadType" on the left is improperly aligned. The attribute "aShort" starts on an 8-bit boundary that is not a multiple of the size of a short (ie a multiple of 16). The attribute "aLong" starts on a 24-bit boundary, which is not a multiple of the size of a long (ie 32). The "GoodType" on the right is properly aligned. Even though the attribute "aChar" does not fill up the second 32 bit word, terminal padding is not required by these rules. Padding at the end of the data type is not required unless that form of alignment is needed for structures-within-structures or other forms of aggregation. For example, if the "GoodType" above were to be used as an array element, 8 bits of terminal padding would be required at the end to maintain proper alignment.

## C2 HLA-compliant Track Service Federate

As shown in Section A3, the HLA Track Federate SOM is a combination of the 'Entity-State' object subset of the RPR-FOM, a translation of the Track-IDL and simulation state change interactions from the DICE SOM. The RPR-FOM was chosen in order to explore the use of an international standard and interact with other off-the-shelf HLA compliant simulations and stealth viewers. One decision required was whether to code for RPR-FOM Version 0.7, which many commercial applications utilised at the time, or Version 1.0. Version 1.0 was chosen because of the better translation from the Track IDL and in the belief that the commercial applications will slowly release suitable versions. Development of the 'Track' object within the SOM was done to investigate the reuse between HLA and EXC3ITE. One issue discovered when developing the 'Track' object was alignment rules as mentioned in Section C1. The included RPR-FOM section already followed these alignment rules.

The level of HLA federate compliance achieved by the HLA Track Federate is exactly the same as the DICE HLA PUI.

As the HLA Track Federate is a CORBA (Orbix 3.0) client it is able to make use of naming services to determine the location of servers. This is not a problem when utilising RTI1.3v6 as it does not utilise CORBA internally but it is a problem with RTI1.3NGv2 as it utilises the The ACE ORB (TAO) internally. When the RTI1.3NGv2 HLA Track Federate attempts to create the federation execution it core dumps. This problem was submitted to the HLA Help desk to which the reason for failure given was collision between symbols occurring in libraries supplied by the RTI1.3NGv2 ORB and the CORBA ORB. A possible solution was to rename symbols inside the libraries supplied by the RTI1.3NGv2 ORB. This solution currently has not been explored to the fullest.

So at the moment the RTI1.3v6 version of the HLA Track Federate is able to connect to CORBA servers through IOR files and the Naming Service. The RTI1.3NGv2 version of the HLA Track Federate can only utilise IOR files to connect to CORBA servers.

## C3 A C2 messaging federate

A C2 messaging federate called the HLA-Graphical User Interface (GUI) was developed to utilise C2 simulation with HLA and to reuse the DICE SOM software libraries. The HLA-GUI utilises the DICE SOM and makes full use of its flexibility. Through the GUI a user is able to receive DICEForm sent by nodes from within the DICE simulation. Also the user is able to construct a DICEForm by either entering the full slash de-limited form or building up the DICEForm through filling in of individual sets and fields.

## C4 Federation object model

The resultant FOM is a combination of the DICE SOM and the HLA Track Federate SOM. As both SOMs were developed at the same time no conflict of objects and/interactions occurred when combining them to construct the FOM.

**Experiences in the Development of EXC3ITE-Based HLA-Compliant Simulation Capabilities**

*Mike Davies, Carsten Gabrisch, Karyn Matthew, John M. Dunn*

(DSTO-TR-1147)

DISTRIBUTION LIST

| | | Number of Copies |
|---|---|---|
| **AUSTRALIA** | | |
| **DEFENCE ORGANISATION** | | |
| **Task sponsor:** | | |
| DGC4 | | 1 |
| **S&T Program** | | |
| Chief Defence Scientist | ) | |
| FAS Science Policy | ) | 1 shared copy |
| AS Science Corporate Management | ) | |
| Director General Science Policy Development | ) | |
| Counsellor, Defence Science, London | | Doc Control Sheet |
| Counsellor, Defence Science, Washington | | Doc Control Sheet |
| Scientific Adviser to MRDC Thailand | | Doc Control Sheet |
| Scientific Adviser - Policy and Command | | 1 |
| Navy Scientific Adviser | | 1 copy of Doc Control Sheet and 1 distribution list |
| Scientific Adviser - Army | | Doc Control Sheet and 1 distribution list |
| Air Force Scientific Adviser | | 1 |
| Director Trials | | 1 |
| **Aeronautical & Maritime Research Laboratory** | | |
| Director | | 1 |
| Dr John Best, Maritime Operations Division | | 1 |
| Dr Mather Mason, AOD | | 1 |
| **Electronics and Surveillance Research Laboratory** | | |
| Director | | 1 copy of Doc Control Sheet and 1 distribution list |
| Chief Information Technology Division | | 1 |
| Research Leader Command & Control and Intelligence Systems | | 1 |
| Research Leader Military Information Enterprise | | 1 |
| Research Leader Advanced Computer Capabilities | | Doc Control Sheet |

| | |
|---|---|
| Research Leader Joint Systems | 1 |
| Research Leader Military Systems Experimentation | 5 |
| Head, Information Warfare Studies Group | 1 |
| Head, Enterprise Visualisation, Instrumentation and Synchronisation Group | Doc Control Sheet |
| Head, Trusted Computer Systems Group | Doc Control Sheet |
| Head, Systems Simulation and Assessment Group | 1 |
| Head, C3I Operational Analysis Group | Doc Control Sheet |
| Head, Information Exploitation Group | 1 |
| Head, Intelligence Group | Doc Control Sheet |
| Head, Human Systems Integration Group | Doc Control Sheet |
| Head, C2 Australian Theatre | 1 |
| Head, Distributed Systems Group | 1 |
| Head, C3 Information Systems Concepts Group | 1 |
| Head, Military Systems Synthesis Group | Doc Control Sheet |
| Head, Systems of Systems Group | 1 |
| Head, Advanced Network Integrity Group | Doc Control Sheet |
| Authors | |
| Mike Davies | 1 |
| Carsten Gabrisch | 1 |
| Karyn Matthews | 1 |
| John M. Dunn | 1 |
| Publications & Publicity Officer, ITD /Executive Officer, ITD | 1 shared copy |
| Dr Anthony Finn, EWD | 1 |
| Dr Anne-Marie Grisogono, LOD | 1 |
| Dr Jon Vaughan, LOD | 1 |

**DSTO Library and Archives**

| | |
|---|---|
| Library Fishermans Bend | 1 |
| Library Maribyrnong | Doc Control Sheet |
| Library Salisbury | 2 |
| Australian Archives | 1 |
| Library, MOD, Pyrmont | Doc Control Sheet |
| US Defense Technical Information Center | 2 |
| UK Defence Research Information Centre | 2 |
| Canada Defence Scientific Information Service | 1 |
| NZ Defence Information Centre | 1 |
| National Library of Australia | 1 |

**Capability Systems Staff**

| | |
|---|---|
| Director General Maritime Development | Doc Control Sheet |
| Director General Aerospace Development | Doc Control Sheet |
| Director General Land Development | |

**Knowledge Staff**

| | |
|---|---|
| Director General Command, Control, Communications and Computers (DGC4) Attention: CMDR J. Langsford DDHQ | 1 |
| DGSIM | 1 |
| Director General Intelligence, Surveillance, Reconnaissance and Electronic Warfare (DGISREW) R1-3-A142 Canberra ACT 2600 | Doc Control Sheet |
| Director General Defence Knowledge Improvement Team (DGDKNIT) R1-5-A165, Canberra ACT 2600 | Doc Control Sheet |

**Navy**

| | |
|---|---|
| SO (Science), Catherine Morgan, COMAUSNAVSURFGRP, Bldg 95, Garden Island, Locked Bag 12, Pyrmont NSW 2009. | Doc Control Sheet |

**Army**

| | |
|---|---|
| Stuart Schnaars, ABCA Standardisation Officer, Tobruck Barracks, Puckapunyal VIC 3662 | 4 |
| SO (Science), Deployable Joint Force Headquarters (DJFHQ) (L), MILPO, Gallipoli Barracks, Enoggera, QLD 4052 | Doc Control Sheet |
| NPOC QWG Engineer NBCD Combat Development Wing, Tobruk Barracks, Puckapunyal, 3662 (Doc Data Sheet relating to NBCD matters only) | Doc Control Sheet |

**Intelligence Program**

| | |
|---|---|
| DGSTA Defence Intelligence Organisation | 1 |
| Manager, Information Centre, Defence Intelligence Organisation | 1 |

**Corporate Support Program**

| | |
|---|---|
| Library Manager, DLS-Canberra | Doc Control Sheet |

**Universities and Colleges**

| | |
|---|---|
| Australian Defence Force Academy | 1 |
| Library | 1 |
| Head of Aerospace and Mechanical Engineering | 1 |
| Serials Section (M list)), Deakin University Library | 1 |
| Hargrave Library, Monash University | Doc Control Sheet |
| Librarian, Flinders University | 1 |

**Other Organisations**

| | |
|---|---|
| OIC Simulation Section ADFWC | 1 |
| Commander Surveillance & Control Group | 1 |
| NASA (Canberra) | 1 |
| AusInfo | 1 |
| State Library of South Australia | 1 |
| Parliamentary Library, South Australia | 1 |

## OUTSIDE AUSTRALIA

| | |
|---|---|
| **Abstracting and Information Organisations** | |
| Library, Chemical Abstracts Reference Service | 1 |
| Engineering Societies Library, US | 1 |
| Materials Information, Cambridge Scientific Abstracts US | 1 |
| Documents Librarian, The Center for Research Libraries, US | 1 |
| **Information Exchange Agreement Partners** | |
| Acquisitions Unit, Science Reference and Information Service, UK | 1 |
| Library - Exchange Desk, National Institute of Standards and Technology, US | 1 |
| SPARES | 5 |
| **Total number of copies:** | **73** |

Page classification: UNCLASSIFIED

## DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA

| | |
|---|---|
| | 1. PRIVACY MARKING/CAVEAT (OF DOCUMENT) |
| 2. TITLE<br>Experiences in the Development of EXC3ITE-Based HLA-Compliant Simulation Capabilities | 3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION)<br>Document (U)<br>Title (U)<br>Abstract (U) |
| 4. AUTHOR(S)<br>Mike Davies, Carsten Gabrisch, Karyn Matthews, John M. Dunn | 5. CORPORATE AUTHOR<br>Electronics and Surveillance Research Laboratory<br>PO Box 1500<br>Salisbury SA 5108 Australia |

| 6a. DSTO NUMBER | 6b. AR NUMBER | 6c. TYPE OF REPORT | 7. DOCUMENT DATE |
|---|---|---|---|
| DSTO-TR-1147 | AR-011-860 | Technical Report | May 2001 |

| 8. FILE NUMBER | 9. TASK NUMBER | 10. TASK SPONSOR | 11. NO. OF PAGES | 12. NO. OF REFERENCES |
|---|---|---|---|---|
| N9505/19/248 | JNT 99/138 | DGC4 | 78 | 18 |

| 13. DOWNGRADING/DELIMITING INSTRUCTIONS | 14. RELEASE AUTHORITY |
|---|---|
| To be reviewed three years after date of publication | Chief, Information Technology Division |

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, SALISBURY, SA 5108

16. DELIBERATE ANNOUNCEMENT

17. CASUAL ANNOUNCEMENT Yes

18. DEFTEST DESCRIPTORS
Synthetic environments
Command, Control, Communications and Intelligence
Interoperability
Distributed Interactive Simulation
Simulation
Project Takari


19. ABSTRACT

Program Takari's Experimental C3I Technology Environment (EXC3ITE) will be key to the establishment of modelling and simulation architectures, practices and capabilities that are integrated with real C3I systems to support a range of military experimentation. The US High Level Architecture (HLA) is one component of an emerging encapsulation and architectural standard for simulation that has been mandated within the US DoD. EXC3ITE and the HLA share two main principles of interoperability and reuse; they are both middleware-rich. This commonality needs to be exploited in order to make best use of the HLA standard. EXC3ITE-based HLA-compliant simulation capabilities were recently developed employing a combination of Adacel Technologies and Aspect Computing staff. This report describes the capabilities developed and experiences gained and raises issues and recommendations on the way ahead for EXC3ITE-based simulation and synthetic environments.


Page classification: UNCLASSIFIED

TECHNICAL REPORT DSTO-TR-1147 AR-011-860 MAY 2001

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE & TECHNOLOGY ORGANISATION
DSTO

ELECTRONICS AND SURVEILLANCE RESEARCH LABORATORY
PO BOX 1500 SALISBURY SOUTH AUSTRALIA, 5108
AUSTRALIA, TELEPHONE (08) 8259 5555